**UNIVERSIDAD TECNOLÓGICA DE PEREIRA**
**FACULTAD DE INGENIERÍAS**
**PROGRAMA DE INGENIERÍA ELÉCTRICA**

# "PMITO – PLATAFORMA MÓVIL PARA INVESTIGACIÓN DE TÉCNICAS DE ODOMETRÍA"

AUTORES:

JULIO CESAR YEPES VALENCIA
MARLON ANDREY FERNANDEZ MESA

PEREIRA
2016

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA**
**FACULTAD DE INGENIERÍAS**
**PROGRAMA DE INGENIERÍA ELÉCTRICA**

# "PMITO – PLATAFORMA MÓVIL PARA INVESTIGACIÓN DE TÉCNICAS DE ODOMETRÍA"

AUTORES:

JULIO CESAR YEPES VALENCIA
MARLON ANDREY FERNANDEZ MESA

Proyecto de grado, requisito parcial para optar al título de:
Ingeniero Electricista.

DIRECTOR
Ph.D. LUIS HERNANDO RÍOS G.

PEREIRA
2016

## DEDICATORIA

Este trabajo va dedicado a mis padres Carlos y Elizabeth quienes con sus esfuerzos y trabajo hicieron que este gran paso de mi vida fuese posible. A mi familia por parte de mi madre que siempre estuvieron pendientes de mi.

**Marlon Andrey Fernández**

Este trabajo va dedicado a mis padres Hernán Yepes y Luz Dary Valencia por apoyarme incondicionalmente en el camino de mis estudios, por su preocupación, consejos y sacrificios que permitieron culminar esta etapa de mi vida, a mis familiares y amigos con quienes compartí experiencias a lo largo de mi formación que de una u otra forma me han ayudado a seguir adelante y no desistir en el proceso.

**Julio Cesar Yepes Valencia**

## AGRADECIMIENTOS

En primera instancia agradecemos a nuestro tutor Luis Hernando Ríos por su confianza y colaboración durante el proceso de aprendizaje e implementación de la plataforma móvil PMITO.

Agradecemos al ingeniero Debbie Johan Arredondo y al ingeniero Diego Fernando Zamora por sus accesorias y herramientas brindadas que se implementaron en el proyecto siendo clave para el desarrollo del mismo; a los compañeros y profesores de la sala de investigación de automática y robótica que nos ayudaron con aclaración de dudas y con el espacio de trabajo, a nuestras familias y amigos por todo el apoyo y confianza depositada en nosotros y por último y no menos importante a los profesores de la facultad de ingeniería eléctrica y a la Universidad Tecnológica de Pereira por brindarnos las bases para nuestra formación como profesionales.

## ÍNDICE DE CONTENIDOS

## ÍNDICE DE FIGURAS

## INDICE DE DIAGRAMAS



## INDICE DE GRÁFICAS



## ÍNDICE DE TABLAS

## INDICE DE ANEXOS

# 1. INTRODUCCIÓN

El presente trabajo describe el diseño y construcción de una plataforma móvil con anillos de sensores infrarrojos y sensores ultrasónicos como dispositivos para la medición de distancia, los cuales serán posteriormente utilizados para la navegación de robots móviles en ambientes dinámicos.

Se muestra una breve reseña sobre la arquitectura de robots móviles y la importancia de determinar una estructura de control para formular alternativas de navegación y construcción de mapas de entorno con la implementación de sensores que permitan al sistema conocer la distancia que ha recorrido, su ubicación, y la distancia existente entre el entorno y el sistema mismo, utilizando sistemas embebidos, en este caso tarjetas de la serie ARDUINO.

Fusionar la información que entregan los distintos tipos de sensores al hardware de ARDUINO es uno de los temas a tratar, pues en cada tipo de sensor se usaron técnicas de adquisición de datos que van desde medición de ancho de pulso hasta protocolo de comunicación serial RS-232 e I2C, con el debido procesamiento de cada fuente de datos en el sistema embebido.

Un preámbulo sobre el trabajo realizado, está ligado directamente a la naturaleza inherente de los sistemas embebidos para ejecutar tareas en concurrencia y el proceso para la medición de distancia, con los datos que entregan los bloques de percepción infrarroja y los otros bloques sensoriales.

Por ejemplo, en esta aplicación los sistemas embebidos están en capacidad de capturar datos de determinado número de fuentes de datos, estos, están limitados sólo por el número de entradas de la misma; el proceso de medición de distancia en sistemas embebidos está implementado para la navegación y construcción de mapas de entorno usando el computador como interfaz gráfica, para determinar la ruta de navegación.

## 1.1 PLANTEAMIENTO DEL PROBLEMA

El evadir obstáculos y navegar por entornos dinámicos es uno de los objetivos de los robots móviles autónomos; en los actuales momentos hacer que los robots móviles se desplacen por entornos dinámicos con un alto grado de precisión es un problema que aún no está resuelto en su totalidad.

Una de las cualidades de los robots móviles autónomos es la precisión con que estos perciben el entorno por donde se desplazan; de dicha precisión se puede garantizar una buena interacción del robot móvil con su entorno de trabajo.

En la bibliografía se referencian robots móviles autónomos en entornos de trabajo transportando materiales y herramientas e interactuando con las personas para cumplir tareas específicas. Para la realización de dichas tareas se requiere la implementación de sofisticados sistemas tanto de control como de instrumentación.

La pregunta objeto de Investigación se podría definir: ¿Será posible a través de la implementación de un bloque de percepción con sensores, poder definir con precisión la posición y orientación de la plataforma móvil en cualquier instante de tiempo?

Con esta tesis se pretende diseñar y construir una plataforma para la navegación en entornos dinámicos implementando técnicas de odometría para definir la posición y orientación correcta de la plataforma en su entorno de trabajo. En cualquier instante de tiempo $t_i$.

# 2. OBJETIVOS

## 2.1 Objetivo general

Diseñar y construir una plataforma móvil para el estudio de técnicas de odometría.

## 2.2 Objetivos específicos

- Diseñar y construir el sistema de locomoción de la plataforma móvil.
- Diseñar y construir el bloque de percepción sensorial compuesto por un anillo de sensores ultrasónicos y un anillo de sensores infrarrojos.
- Diseñar la lógica de control de la locomoción de la plataforma móvil y del bloque de percepción utilizando el hardware libre ARDUINO MEGA 2560.
- Implementar algoritmos de navegación y localización.

## 3. DISEÑO METODOLÓGICO

Con el fin de responder al carácter investigativo del proyecto, alcanzar los objetivos y comprobar la veracidad del planteamiento del mismo, se tendrá en cuenta una metodología teórico - experimental implementada en cada uno de las etapas del proyecto, además de la utilización de la simulación y experimentación para detectar y resolver problemas técnicos que puedan dificultar el logro de los objetivos.

El proyecto de investigación se llevará a cabo en fases que delimitan y especifican cada una de las actividades a desarrollar.

• *Fase I: "Estudio de la técnica a emplear".* En esta fase se desarrollará una recopilación de información, estudio de material académico, libros, artículos especializados para obtener las primeras aproximaciones a las implementaciones deseadas.

• *Fase II: "Construcción de la Plataforma Móvil".* En esta fase se construirá la plataforma móvil y se adaptarán todos los módulos que la componen (motores, encoders, sensores, brújula) para tener un adecuado desempeño de la misma.

• *Fase III: "Implementación de técnicas de odometría".* En esta etapa se diseñarán e implementarán las técnicas necesarias para lograr los objetivos propuestos. Al tener los conceptos teóricos de la fase I, se pueden resolver problemas y generar distintas combinaciones de las técnicas propuestas para lograr el mejor desempeño.

• *Fase IV: "Pruebas en la Plataforma Móvil".* En esta fase se utiliza la plataforma móvil para realizar pruebas a todos los módulos implementados. Se diseñará una interfaz gráfica para tomar datos del entorno y definir la posición y orientación de la plataforma móvil en cualquier instante $t_i$.

• *Fase V: "Presentación de Resultados"*. En esta última fase se presentará el documento final a la comunidad en general.

## 4. ESTADO DEL ARTE

### 4.1. Arquitectura del robot.

Los primeros trabajos en robótica móvil fueron para ambientes industriales con un grado mínimo de incertidumbre, por lo cual la capacidad sensorial era la menor, de tal forma que los robots móviles necesitaban de los sensores sólo para la realización de tareas muy específicas [1].

En los últimos años el principal objetivo de investigación en el área de la robótica móvil es la construcción de entes o sistemas que tomen la información proporcionada por un módulo sensorial, como sensores de ultrasonidos, infrarrojos, visión artificial, entre otros., utilizando técnicas para moverse inteligentemente por un entorno y realizar una serie de tareas de forma autónoma.

"Un robot móvil debe estar en condición de procesar la información sensorial suministrada por los sensores y adoptar una acción de control que le lleve a un estado definido como meta" [2].

Actualmente el desarrollo la navegación de los sistemas robóticos, es el resultado del trabajo conjunto de especialistas en áreas como ingeniería mecánica, ingeniería de sistemas, telemática, arquitectura de computadores, ingeniería del software e inteligencia artificial.

Atendiendo a la recomendación general que la *IEEE Robotics and Automatización Soviet* lanzó en el año 2001 [3], el desafío actual es encontrar las aplicaciones adecuadas para los desarrollos y aumentar las capacidades de los sistemas robóticos para extenderlos a nuevas aplicaciones.

La tendencia de los diferentes grupos de investigación en el área de robótica se orienta a que un robot móvil esté capacitado para construir un mapa de su entorno de trabajo, así como de navegar de manera autónoma, evitando obstáculos a través de él.

Una situación más deseable y más desafiante es que el robot móvil tenga la capacidad de generar y mantener su mapa de manera autónoma, mientras logra los objetivos propuestos.

Son muchas las herramientas que se han implementado en el logro de estos objetivos, una de las estrategias propuestas se basa en el empleo de las técnicas inteligentes y la navegación basada en comportamientos.

El desarrollo de robots móviles capaces de construir su propio mapa y navegar de manera autónoma, requiere de estrategias de control inteligente, que puedan manejar la incertidumbre presentada por el entorno de trabajo, mientras se desempeñan en tiempo real con una relativa baja carga computacional.

### 4.2. Reto de la navegación autónoma

Cualquier aproximación al control de un sistema dinámico necesita utilizar algún conocimiento o modelo del sistema a controlar.

En el caso de un robot móvil, este sistema consiste del robot en sí, más el ambiente en el que opera. Desafortunadamente, mientras el modelo del robot como tal puede ser obtenido, la situación es diferente si éste se considera dentro de cualquier tipo de ambiente real y sin ningún arreglo previo del entorno.

Los ambientes están caracterizados por la presencia de incertidumbre; la naturaleza de los fenómenos involucrados es tal, que frecuentemente no se está en capacidad de precisar un modelo o cuantificar estos fenómenos. Por ejemplo, si se considera la incertidumbre inducida por la presencia de personas en un ambiente. Las personas se mueven alrededor y pueden cambiar la posición de los objetos y los enseres del entorno; sin embargo, en la mayoría de los casos no se puede obtener una distribución probabilística que pueda caracterizar estos eventos. La interacción del robot con el ambiente causa dificultades similares y sus acciones de sensado son influenciados por condiciones ambientales, las cuales son difícilmente contabilizadas; por ejemplo, el error en el movimiento del robot puede variar como resultado de un piso mojado, la calidad del reconocimiento visual puede ser afectado por un rápido cambio de las condiciones de luz.

Una estrategia para afrontar esta gran cantidad de incertidumbre es abandonar la idea de obtener un modelo completo del ambiente en la fase de diseño y dotar al robot con la capacidad de construir este modelo por sí mismo y en línea. Esta estrategia es llamada Arquitectura Jerárquica [4], como lo muestra la Figura 1.

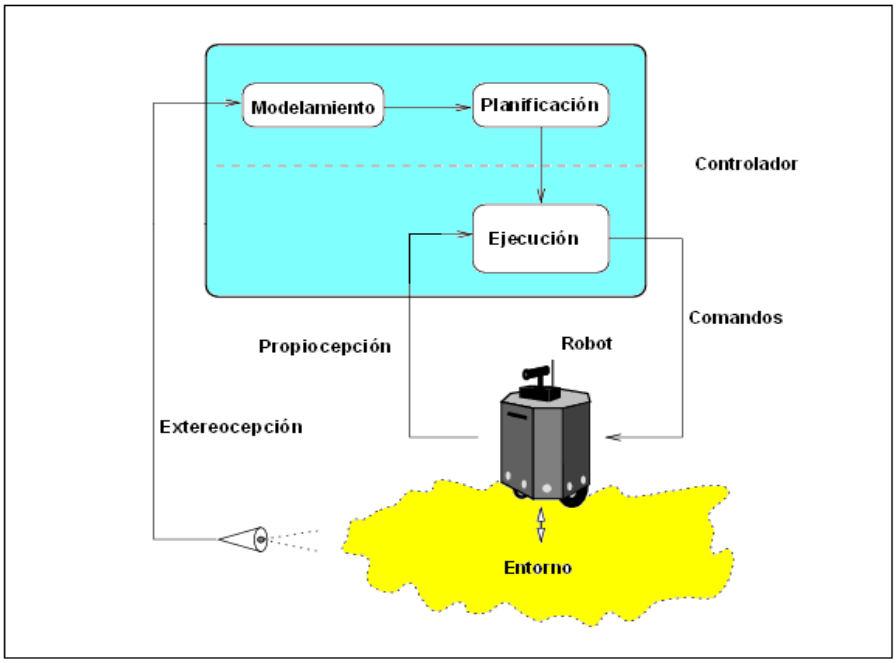



Figura 1. Arquitectura Jerárquica.

Utilizando los sensores de posición, el robot adquiere un modelo del espacio de trabajo en el mismo instante en que la tarea será realizada. En este modelo se basa un programa planeador para construir un plan que ejecutará la tarea dada en el ambiente dado. Este plan es pasado al programa de control de bajo nivel.

Típicamente el procedimiento de ejecución es ciego. El controlador puede utilizar un modelo del robot y monitorear el estado de los efectores del robot, pero esto no permite sensar o modelar el ambiente una vez más.

No es difícil observar las limitaciones de la Arquitectura Jerárquica para trabajo en tiempo real en ambientes de trabajo. El modelo adquirido por el robot necesariamente es incompleto e inexacto debido a la incertidumbre en la percepción, además el modelo se desactualiza rápidamente por la dinámica del ambiente entonces el plan construido puede

ser inadecuado para el ambiente encontrado durante la fase de ejecución, otro factor preponderante es que los procesos de modelado y planeamiento son computacionalmente complejos y consumen mucho tiempo de cómputo, con lo cual se agranda el problema; intuitivamente una realimentación (lazo de realimentación) con el ambiente puede mejorar estas limitaciones, es por ello que aparece el enfoque SMPA *"SENSE – MODEL – PLAN - ACT*". La complejidad del SMPA puede hacer que la respuesta temporal del sistema en ambientes dinámicos tarde incluso segundos.

Las investigaciones en robótica móvil tienden al desarrollo de numerosas y nuevas arquitecturas que integran Percepción y Acción. La tendencia general se orienta al planeamiento de pequeñas suposiciones sobre el ambiente encontrado en la fase de ejecución y que la ejecución sea sensible al ambiente y se adapte a las contingencias encontradas.

Para lograr esto, los datos de percepción han sido incluidos en la capa de ejecución (Arquitecturas Híbridas). la cual se observa en la Figura 2.

La aparente extensión del sistema receptor al módulo de ejecución tiene dos consecuencias importantes:

- La interacción del robot con el ambiente es mucho más ceñida dado que ahora el ambiente está incluido como lazo cerrado con la capa o módulo de ejecución.
- La complejidad de la capa de ejecución ha sido incrementada debido a la necesidad de considerar múltiples objetivos: alcanzar los objetivos tácticos tomados del planeador (capa o fase de planificación) y reaccionar ante los eventos ambientales detectados por la percepción.

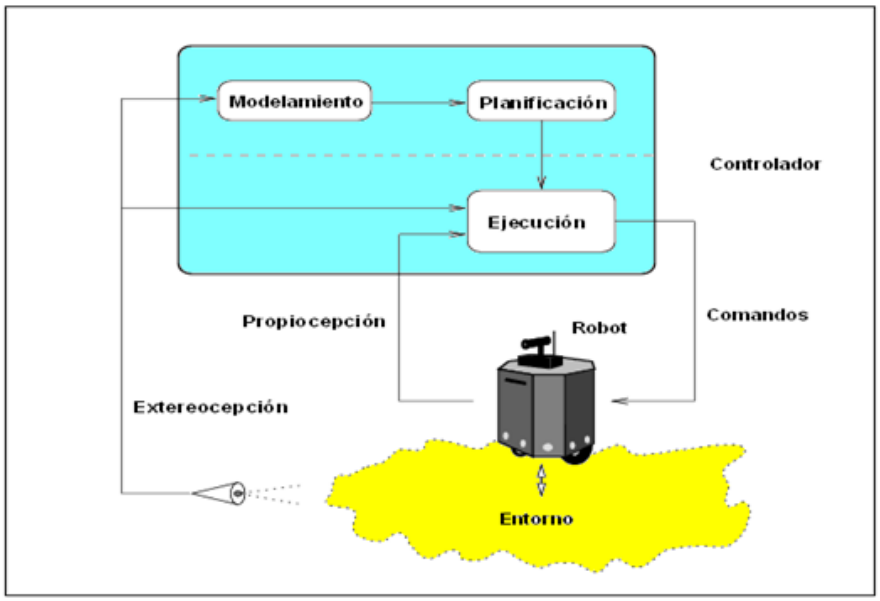



Figura 2. Arquitectura Híbrida.

Muchos investigadores han escogido cubrir esta complejidad con la estrategia "Divide y Conquistarás", pero esto ha abocado a diferentes estilos de división. Algunos han perseverado dentro de la tradición jerárquica y tienen fuertemente seccionada la capa de ejecución en capas de secuencias y capas de proceso. Otros como Mataric [5] y Parker [6] parten de la Arquitectura Jerárquica y descomponen la capa de ejecución en pequeñas unidades de Decisión-Acción o comportamientos (Arquitectura Híbrida Basada en Comportamientos).

Un comportamiento es una pequeña unidad de control hecha para alcanzar un objetivo simple en un restringido conjunto de situaciones. Para poder realizar tareas complejas los robots autónomos necesitan, de la activación y cooperación de un número de comportamientos, cada comportamiento implementa en detalle un control seguro, para una sub-tarea específica como seguir una trayectoria, evitar objetos sensados o cruzar puertas.

Las Arquitecturas Híbridas no resuelven el problema de navegación autónoma pero ellas proporcionan una estructura conveniente en la cual los diferentes sub-problemas pueden ser repartidos e integrados.

### 4.3. Línea de acción

La implementación de un tipo de arquitectura determinado para el proyecto está ligada directamente a la acción en tiempo real, y la capacidad de ejecutar tareas. Si se piensa en el ser humano desde un punto de vista perceptivo, éste se denota como un ente dotado de una cantidad ilimitada de sensores en funcionamiento y en concurrencia, con una capacidad de organizar las variables entregadas por los mismos y decidir cuál de estas variables tiene un valor ponderado que requieran una acción de control. Por esto, la implementación de hardware programable encaja perfectamente en el perfil para generar sistemas con un grado de autonomía mayor, pues con estos dispositivos la ejecución de procesos en concurrencia es algo inherente, además que los procesos pueden adquirir un valor ponderado como resultado de un orden jerárquico que es asignado por el diseñador, sin necesidad de tener en cuenta conceptos utilizados en sistemas anteriores como multiplexar datos y secuencias serie que aportan más tiempo en ejecución de cómputo.

### 4.4 Arquitecturas de control en robótica móvil.

Una arquitectura para robot móvil se compone tanto del software como del hardware involucrado en el control del mismo. Los robots son sistemas complejos difíciles de desarrollar. Integran múltiples sensores y actuadores, tienen muchos grados de libertad y deben tratar con tareas de tiempo real. Sobre los elementos que debe poseer un robot, R. R. Murphy dice [7]:

En Inteligencia Artificial ha quedado establecido que cualquier robot autónomo tiene que poseer las siguientes tres primitivas o elementos funcionales:

a. SISTEMA DE PERCEPCIÓN: Capacita al robot para disponer de la información de los sensores y producir, a partir de ésta, información útil para los otros elementos funcionales.

b. SISTEMA DE RAZONAMIENTO O PLANIFICACIÓN: Capacita al robot para producir las tareas a ejecutar (ir a la entrada, girar a la derecha, moverse 2 metros y parar, etc.) en base a la información sensorial y conocimiento disponible del mundo que le rodea o entorno.

c. SISTEMA DE EJECUCIÓN O ACCIÓN**:** Capacita al robot para la ejecución de tareas mediante la interacción con los actuadores.

# 5. DESARROLLO DE LA PLATAFORMA ROBÓTICA PMITO

En robótica móvil se usan una gran variedad de sensores que miden distintas magnitudes. Los que más se utilizan son los orientados a resolver uno de los problemas fundamentales de todo robot móvil: la determinación de la posición. En los trabajos de Everett y Peters [8], y Feng, Borenstein y Everett [9], se lleva a cabo una revisión de todas las tecnologías y metodologías utilizadas a nivel de sensores y procesamiento de la información proporcionada por los mismos para resolver el problema de la localización de un robot. Adams [10], tiene otra revisión algo más actualizada que incluye detalles sobre el procesamiento básico de cada sensor. Básicamente el problema de la localización ó el posicionamiento se resuelve mediante medidas relativas de posición o mediante sistemas absolutos de posicionamiento.

Para el desarrollo del proyecto se utilizaron en la sensorización de la plataforma móvil (Robot Móvil):

- Sensores ultrasónicos para medir distancias largas de obstáculos.
- Sensores infrarrojos para medir distancias cortas de obstáculos.
- Encoders, como medidores de distancia recorrida por las ruedas (sensores odometricos).
- Brújula digital, como sensor de dirección y rumbo.

## 5.1. Sensor ultrasónico (HC-SR04)

El robot incluye un anillo sensorial que se compone de ocho sensores de ultrasonido ubicados en la parte superior de la plataforma (véase la figura 9), tres de ellos son colocados en la parte delantera, uno en la parte trasera y los otros cuatro son colocados de a dos en la partes laterales, este dispositivo sirve para detectar los objetos que se encuentren alrededor del anillo midiendo la distancia y así poder tomar decisiones para esquivar el objeto sin dificultades. En la figura 3, se puede observar el dispositivo de ultrasonido.


Figura 3. Sensor de ultrasonido

Fuente: Luis llama. *Ingeniería, informática y diseño* [en línea]. Disponible en web: <http://www.luisllamas.es/2015/06/medir-distancia-con-arduino-y-sensor-de-ultrasonidos-c-sr04/ >

Este sensor funciona mediante la transmisión de una ráfaga de ultrasonido en una frecuencia muy por encima del rango auditivo humano y provee un pulso de salida, el cual corresponde con el tiempo requerido por el eco (rebote) para retornar hasta el sensor. Al medir la duración de este pulso se puede calcular fácilmente la distancia al objetivo retorno.

Dónde:

$$Distancia = Velocidad\ del\ sonido * fact * Tiempo \quad \text{(Ec .1)}$$

$V_S$=Velocidad del sonido

$V_S = 343m/s$ o $V_S = 0.0343cm/\mu s$

$fact$=0.5 la multiplicación por este factor es por el recorrido de la onda (ida y vuelta).

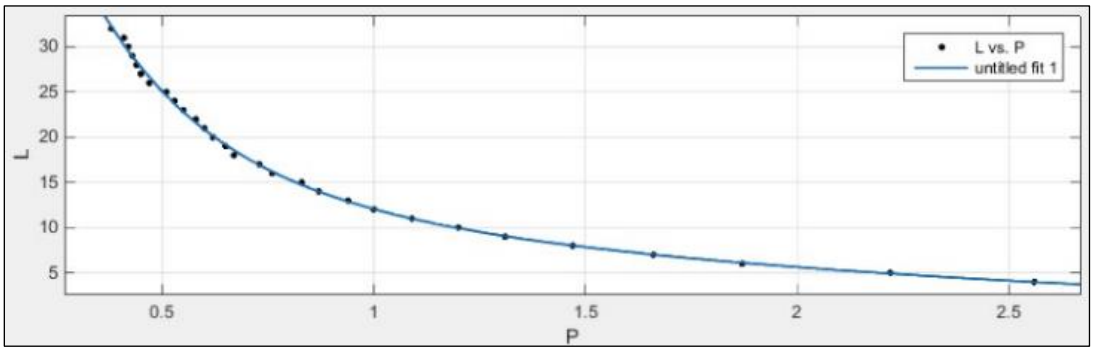
### 5.2. Sensor infrarrojo Sharp (2Y0A21)

El robot incluye un anillo sensorial que está ubicado en la parte inferior de la plataforma (véase la figura 9), la cual se compone de ocho sensores infrarrojos, tres de ellos son colocados en la parte delantera, uno en la parte trasera y los otros cuatro son colocados de a dos en la parte lateral, este dispositivo sirve para detectar los objetos que se encuentren alrededor del anillo para un rango de distancia entre 0 y 20 cm.

Figura 4. Sensor infrarrojo

El sensor infrarrojo (ver figura 4) se basa en la intensidad de una señal luminosa. La luz se puede definir como el conjunto de radiaciones electromagnéticas que forman parte del espectro visible. La radiación se puede definir como la emisión o transferencia de energía en forma de ondas electromagnéticas o partículas. Para el funcionamiento del dispositivo debe ser excitado el emisor con un pulso de corriente continua, emitiéndose así un haz de luz del espectro infrarrojo que al chocar con algún obstáculo es captado por el fotodiodo el cual produce un pequeño voltaje de salida que depende de la cantidad de energía que alcance al receptor. La energía que refleja el objeto depende principalmente de la distancia a la que se encuentre, del coeficiente de reflexión del objeto (dependiente a su vez del color del objeto, del brillo y de la textura).

Para la calibración del sensor infrarrojo, se realizó un programa en MATLAB que se puede ver en el ANEXO 1 donde se obtiene la gráfica de calibración para cada sensor infrarrojo. En la gráfica 1. se puede observar la curva característica del infrarrojo.




Gráfica 1. Característica de la distancia del objeto

Donde

L=Distancia al objeto reflectante (cm).

P=Salida análoga de voltaje (v).

$f(x)$=Distancia del objeto

La función $f(x)$ es una exponencial que relaciona la (grafica 10.) con las distancias reflectante del objeto.

$$f(x) = a * \exp(b * x) + c * \exp(\mathrm{d} * \mathrm{x}) \quad \text{(Ec .2)}$$

$a = 69.12 \qquad b = -3.622 \qquad c = 18.65 \qquad d = -0.6034$

### 5.3 Encoders stemann

El encoders stemann (ver figura 5) es un transductor rotativo que transforma un movimiento angular en una serie de pulsos digitales. Estos pulsos generados pueden ser utilizados para controlar los desplazamientos de tipo angular o de tipo lineal.

Las señales eléctricas de rotación pueden ser elaboradas mediante controles numéricos (CNC), contadores lógicos programables (PLC), sistemas de control etc. Las aplicaciones principales de estos transductores están en las máquinas herramienta o de elaboración de materiales, en los sistemas de motores, en los aparatos de medición, control y en los robots como es nuestro caso.

El sistema de lectura se basa en la rotación de un disco grabado con un reticulado radial formado por líneas opacas, alternadas con espacios transparentes. Este conjunto está iluminado de modo perpendicular por una fuente de rayos infrarrojos. El disco proyecta de este modo su imagen sobre la superficie de varios receptores oportunamente enmascarados por otro reticulado que tiene el mismo paso del anterior llamado colimador.

Los receptores tienen la tarea de detectar las variaciones de luz que se producen con el desplazamiento del disco convirtiéndolas en las correspondientes variaciones eléctricas.

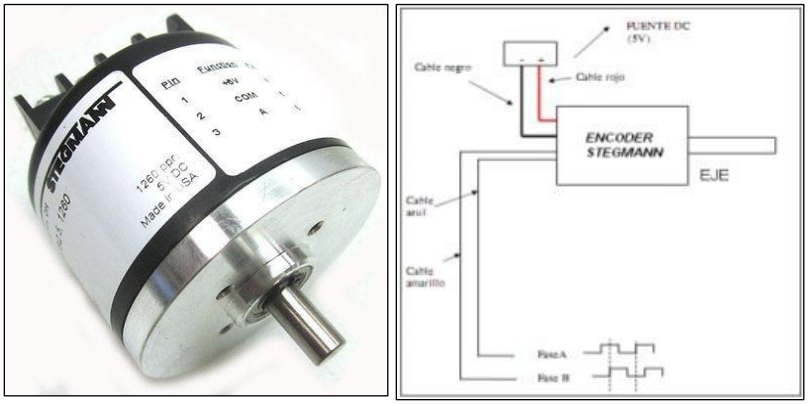



Figura 5. Encoders stemann.

### 5.4 Brújula digital (hmc5883l)

El módulo HCM5883L (ver figura 6) es un dispositivo capaz de detectar los campos magnéticos terrestres, Esta tecnología fue desarrollada por Honeywell.
Su funcionamiento se lleva a cabo mediante unos sensores magneto resistivos que aplicando una diferencia de potencial los sensores convierten cualquier campo magnético incidente en la dirección del eje sensible con una salida de voltaje diferencial. Estos sensores están hechos de un hierro de níquel de película delgada y modelado como un elemento de tira resistiva. Cuando hay presencia de un campo magnético, un cambio en los elementos resistivos puente provoca un cambio correspondiente en la tensión a través de las salidas.


Figura 6. Brújula digital (hmc5883l).

Es un dispositivo capaz de detectar el campo magnético terrestre en los ejes x, y, z. Tendremos que tener en cuenta que la brújula es sensible a los campos magnéticos que la rodea, por lo tanto si tenemos metales o dispositivos cerca puede influir en el comportamiento del chip. Usaremos la brújula para detectar el campo magnético terrestre y así saber dónde se encuentra el Sur magnético y por consiguiente saber dónde está el Norte el Este y el Oeste. También hay que tener en cuenta que el sensor detecta los campos magnéticos en el eje z, por lo tanto lo tenemos que colocar lo más paralelo posible al suelo para obtener una mayor precisión.

Después del apartado teórico vamos a la práctica. Usaremos dos resistencias de 2.2k tal y como recomienda el datasheet del sensor entre los pines Vcc-SDA y Vcc-SCL. El pin Vcc deberá ir a la entrada de 3.3V de nuestro Arduino ya que el sensor necesita un voltaje entre 2.16 a 3.6. Ya que el sensor funciona mediante I2C, estos es, línea de datos SDA y línea de reloj SCL tenemos que usar los Pines 20 y 21 de nuestro Arduino mega respectivamente.

## 5.5 Diseño mecánico

Este capítulo abarca los temas de diseño de las partes mecánicas de la plataforma, incluyendo los materiales, formas de configuración, dimensiones. Además el diseño requiere de la adecuación de dos anillos de percepción sensorial que tengan como función tomar lecturas de distancia en todo momento.

### 5.5.1. Modelo de locomoción

Con el estudio realizado en locomoción para el desarrollo del proyecto se elige la configuración de disposición de ruedas con guiado diferencial debido a que se facilita su diseño y construcción donde se puede tener un gran control de los movimientos, además el error odométrico visto en trabajos anteriores es óptimo para la aplicación.

La configuración del sistema está diseñada para dos ruedas con tracción independiente, (véase figura 7) donde cada rueda está acoplada a un motor, además se le colocaron dos ruedas “locas” para mantener la dirección del robot. Estas ruedas no llevan asociadas ningún motor, giran libremente.

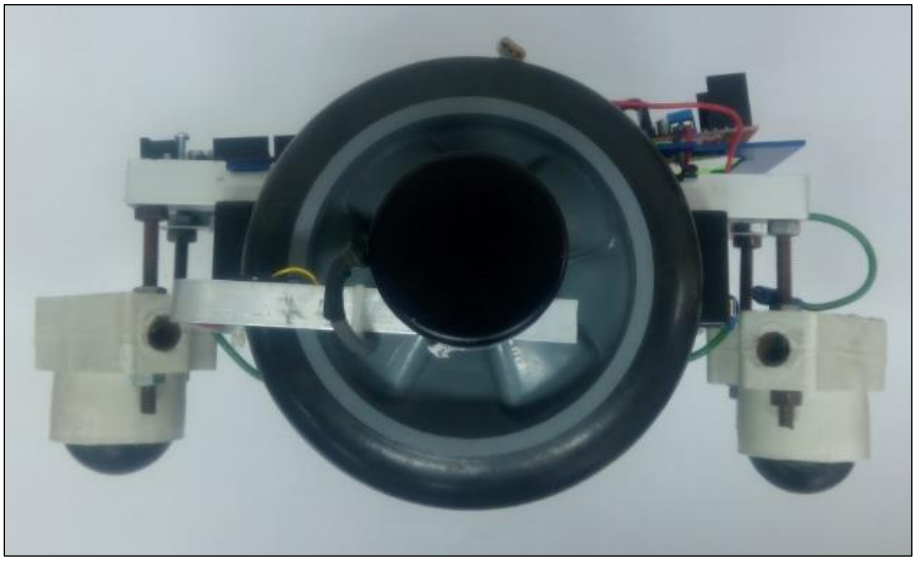


Figura 7. Configuración diferencial.

### 5.5.2. Chasis

El chasis del robot es una de las partes principales del diseño mecánico, ya que el diseño debe contar con una estructura central capaz de soportar todos los componentes necesarios para su autonomía, si el chasis no es el adecuado tendríamos serios problema porque el robot no podrá desplazarse en la dirección asignada. El diseño del chasis está dividido en dos partes principales que conforman el cuerpo del robot.

La primer parte del chasis está ubicada en la parte inferior de la plataforma donde se puede observar en la figura 8, cómo se van a ubicar las bacterias, los motores y los encoders . La construcción de la base de “PMITO” fue hecha en aluminio de referencia P5 para poder mantener un buen equilibrio con el peso.

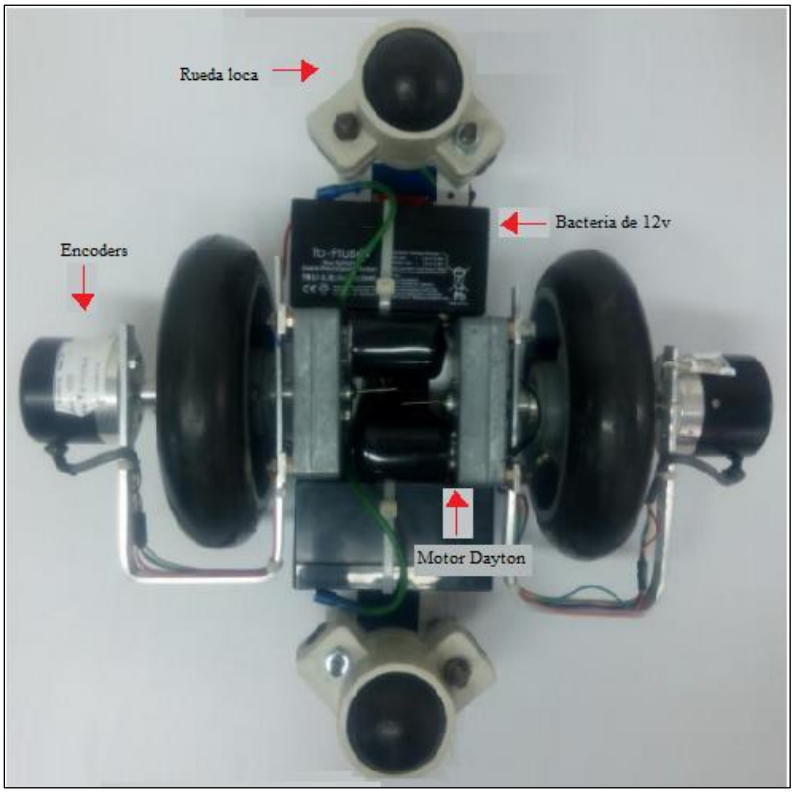



Figura 8. Foto de la parte inferior del chasis.

La segunda parte del chasis está ubicada en la parte superior del cuerpo robot donde se puede observar en la figura 9. Esta parte es diseñada para mapear el entorno y tomar medición de distancias a la cual se encuentra el objeto de los sensores.

La ubicación de los sensores se realizó de forma de octágono con la estrategia de realizar lectura secuencial para no tener errores o conflictos con la medición de cada sensor .Se puede observar que el interior del robot se encuentran toda la circuitería con el arduino mega y los dos anillos sensoriales conformados por sensores ultrasónicos e infrarrojos .

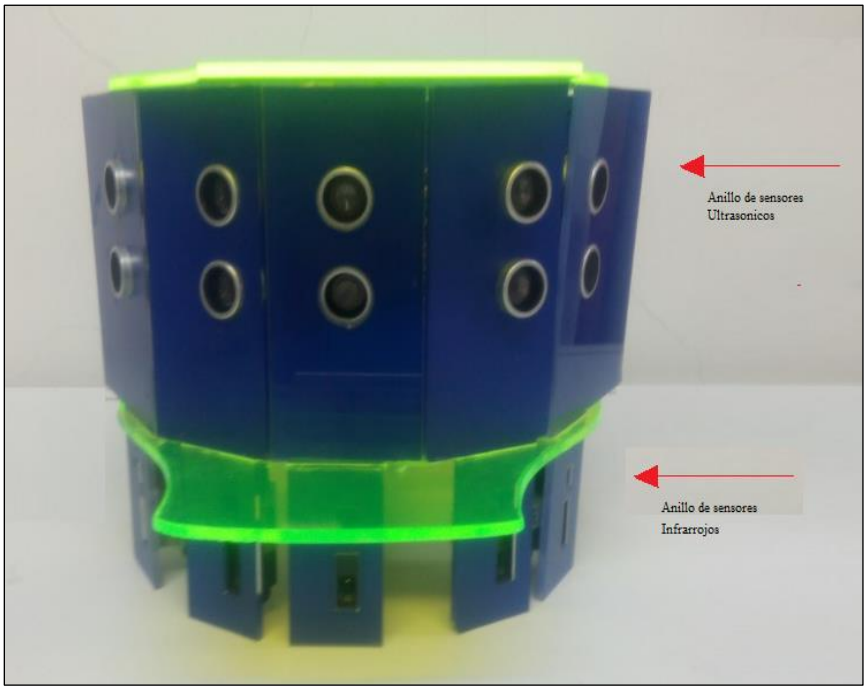



Figura 9. Foto de la parte superior del chasis

### 5.5.3. Configuración de los sensores.

Para conseguir un buen mapeo del entorno del robot, es necesario ubicar los sensores en posiciones estratégicas para captar la distancia de los objetos cercanos en todas las direcciones, evitando puntos ciegos que puedan generar errores de navegación. En las figuras 10 y 11 se ve la distribución de los sensores para cada anillo.

Las lecturas de los ultrasonidos en el anillo se realizaron de manera que no tuviera problema con el crosstalk, la solución fue colocando los ultrasonidos a sensar por grupos, el primer grupo lo conforman los ultrasonidos 1 y 5, el segundo grupo lo conforma el ultrasonido 2, el tercer grupo lo conforman los ultrasonidos 3 y 6, el cuarto grupo lo conforman los ultrasonidos 4 y 7, el último grupo lo conforma el ultrasonido 8.

Para el caso de las lecturas de los sensores infrarrojos se realizó la misma solución por grupos mencionada en el párrafo anterior.

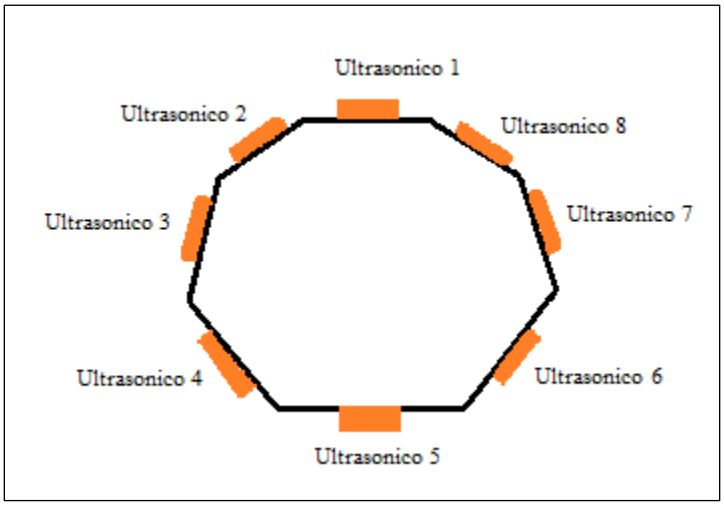



Figura 10. Anillos de sensores Ultrasónicos.

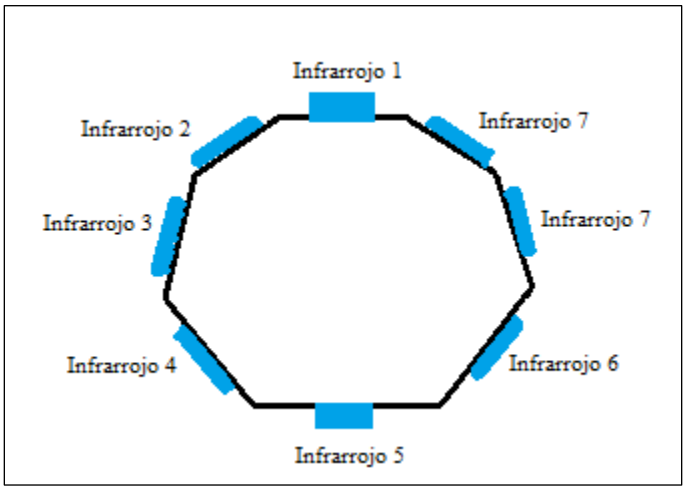



Figura 11. Anillos de sensores Infrarrojos.

### 5.5.4. Sistema de alimentación

Para elegir la alimentación se tuvo en cuenta que la plataforma debería tener dispositivos independientes ya que el sistema no debe de estar enchufado a la red eléctrica debido a que el robot debe de estar en movimiento durante un tiempo prolongado en cualquier entorno.

La bacteria del robot PMITO es una “Tb-Plus TB12-1.3”, (véase figura 12), proporciona 12v con capacidad de 1.3AH.


Figura 12. Bacteria Tb-Plus TB12-1.3

Esta bacteria se escogió por tres factores importantes para el diseño de la plataforma.

- ➢ Bajo costo.
- ➢ Posibilidad de recargar.
- ➢ Fácil de integrar a la plataforma.

### 5.5.5. Motores DC

Los motores DC de marca Dayton (ver la figura 13) son de gran utilidad cuando necesitamos controlar velocidades y además entrega un par relativamente grande a velocidades reducidas.

Este tipo de motores son alimentados a 12v, los cuales desarrolla una velocidad máxima de 18 metros por minuto y se pueden cambiar la dirección de giro invirtiendo su polaridad, su característica de torque es de 20 kg con máxima carga necesarios para obtener un buen desplazamiento.

Figura 13. Motor DC Dayton.

Fuente: Drillspot. *Industrial grade at work* [en línea]. Disponible en web: <http://www.drillspot.com/products/60067/Dayton_2L010_Parallel_Shaft_12_Vdc_Gearor .

### 5.5.6. Driver motor L298N

El módulo para el manejo de los motores DC tanto para su velocidad como su dirección, son controlados mediante el Driver L298N (ver figura 14) que permite el control PWM de los dos motores hasta 2 amperios. Cuenta con jumper de selección para habilitar cada una de las salidas del módulo (A y B). La salida A está conformada por OUT1 y OUT2 que corresponde a los pines del motor A y la salida B por OUT3 y OUT4. Los pines de habilitación son ENA y ENB respectivamente. El módulo es conectado a una bacteria de 12v y a los 5v del arduino mega.

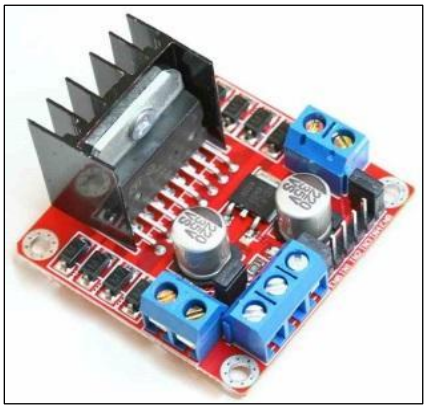


Figura 14. Driver motor L298n.

### 5.6. Diseño electrónico

El diseño electrónico es la parte fundamental del sistema robótico ya que este incluye los principales dispositivos programables y de comunicación para la adquisición de datos y poder llevarlo a una computadora.

#### 5.6.1. Elección de microcontroladores

Para la elección del microcontrolador es necesario estudiar con cuantos pines y cuanta potencia se necesita para un control del robot.

En la parte de inferior de la plataforma teniendo en cuenta que se va a trabajar con dos motores DC que tienen de a dos pines, que son el de 12v y GND. Además tiene que usar el driver del motor y los dos encoders, estos últimos están dotados de cuatro pines, dos de control (A y B) y los pines alimentación ( 5v y GND) correspondiente, estos elementos van conectados a un arduino mega con la configuración de modo esclavo, la transmisión y recepción de los datos se va a manejar con la comunicación maestro esclavo.

Se hace un inventario de todos los pines necesarios en el arduino esclavo:

- 2 pines análogos

- 6 pines digitales, 2 con PWM
- 2 pines de 5v
- 1 pin de SDA
- 1 pin SCL

| **Arduino Mega esclavo** | |
|---|---|
| **Pin** | **Descripción** |
| 39 | Pin digital. Conexión con Driver IN1 |
| 45 | Pin digital. Conexión con Driver IN4 |
| GND | Acoplamiento con driver y maestro |
| 21 | Conexión SCL para comunicación I2C |
| 20 | Conexión SDA para comunicación I2C |
| 2 | Interrupciones encoder A |
| 3 | Interrupciones encoder B |
| 8 | Pin digital. Conexión con Driver ENA |
| 9 | Pin digital. Conexión con Driver ENB |
| 5V | Alimentación a la Brújula |

Tabla 1. Arduino Mega esclavo.

Por otro lado en la parte superior del robot se encuentra dos anillos sensoriales y la brújula digital, estos anillos están conformados por ocho sensores ultrasónicos y ocho sensores infrarrojos.

Los sensores ultrasónicos cuentan con cuatro pines, dos de ellos son pines de control digitales y los otros son la alimentación (Echo, Trigger, 5v y GND) y los sensores

infrarrojos cuentan con tres pines, uno de ellos es el pin de la señal análoga y los otros es la alimentación (5v y GNG), teniendo en cuenta los dispositivos a conectar y el número de pines a utilizar optamos entonces por escoger el tipo de microcontrolador arduino mega como maestro.

Se hace un inventario de todos los pines necesarios en el arduino maestro:

- 16 pines digitales
- 2 pines GND
- 1 pin de 5v
- 1 pin de SDA
- 1 pin SCL

| **Arduino Mega maestro** | | | |
|---|---|---|---|
| **Pin** | **Descripción** | **Pin** | **Descripción** |
| 8 | Pin digital. Conexión con Echo 1 | 30 | Pin digital. Conexión con Echo 8 |
| 9 | Pin digital. Conexión con Trigger 1 | 31 | Pin digital. Conexión con Trigger 8 |
| 10 | Pin digital. Conexión con Echo 2 | A8 | Pin análogo. Conexión con Infrarrojo 1 |
| 11 | Pin digital. Conexión con Trigger 2 | A9 | Pin análogo. Conexión con Infrarrojo 2 |
| 12 | Pin digital. Conexión con Echo 3 | A10 | Pin análogo. Conexión con Infrarrojo 3 |
| 13 | Pin digital. Conexión con Trigger 3 | A11 | Pin análogo. Conexión con Infrarrojo 4 |

| | | | |
|---|---|---|---|
| 22 | Pin digital. Conexión con Echo 4 | A12 | Pin análogo. Conexión con Infrarrojo 5 |
| 23 | Pin digital. Conexión con Trigger 4 | A13 | Pin análogo. Conexión con Infrarrojo 6 |
| 24 | Pin digital. Conexión con Echo 5 | A14 | Pin análogo. Conexión con Infrarrojo 7 |
| 25 | Pin digital. Conexión con Trigger 5 | A15 | Pin análogo. Conexión con Infrarrojo 8 |
| 26 | Pin digital. Conexión con Echo 6 | 20 | Conexión SDA para comunicación I2C |
| 27 | Pin digital. Conexión con Trigger 6 | 21 | Conexión SCL para comunicación I2C |
| 28 | Pin digital. Conexión con Echo 7 | GND 1 | Conexión GND para sensores |
| 29 | Pin digital. Conexión con Trigger 7 | GND 2 | Conexión GND para comunicación I2C |
| | | 5V | Conexión de 5v para sensores |

Tabla 2. Arduino Mega maestro.

#### 5.6.1.2. Arduino Mega 2560

El Arduino Mega 2560 (ver la figura 15) es una placa electrónica basada en el chip Atmega 2560. Cuenta con 54 pines digitales de entrada / salida (de los cuales 15 se pueden utilizar como salidas PWM), 16 entradas analógicas, 4 UARTs (hardware puertos serie), un 16 MHz del oscilador de cristal, una conexión USB, un conector de alimentación.

Características de la placa

- Sistema de comunicación I2C en los pines 20(SDA) y 21(SDL):
- Tiene 4 sistema de comunicación serial (UARTs)
- 6 interrupciones externas en los pines 2, 3, 18, 19, 20 y 21

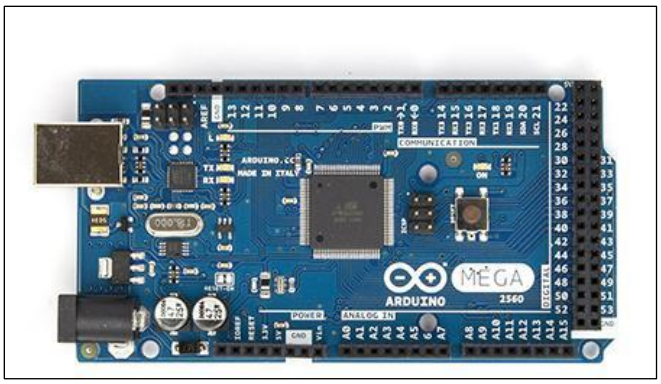



Figura 15 Arduino mega2560.

Fuente: Arduino. *Arduino Mega 2560*[en línea]. Disponible en web: <https://www.arduino.cc/en/uploads/Main/ArduinoMega2560_R3_Front.jpg>

### 5.6.2. Sistema de comunicación Inalámbrica (RF)

#### 5.6.2.1. Módulo Xbee

El módulo Xbee (ver figura 16) es un protocolo de comunicaciones inalámbrico basado en el estándar de comunicaciones para redes inalámbricas IEEE802.15.4. Creado por Zigbee Alliance.

Zigbee permite que dispositivos electrónicos de bajo consumo puedan realizar sus comunicaciones inalámbricas. Es especialmente útil para redes de sensores en entornos industriales, médicos y, sobre todo, domóticos.

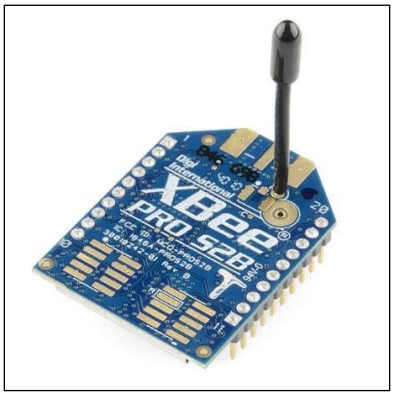



Figura 16. XBEE pro S2.

### 5.6.2.2. Modulo Explorer USB

El Xbee Explorer USB es una tarjeta para los tranceivers Xbee, con la cual se puede programar cualquiera de los módulos Xbee y enviarle datos.

Posee conexión USB directa al computador o al puerto USB sin necesidad de cable.

Trabaja con todos los módulos Xbee incluyendo la serie 1, serie 2. tanto versión standard como versión Pro.

Característica del dispositivo

1. Conversor USB a Serial
2. Conector 2mm para Xbee y conector de 2.54mm para salidas estándar
3. Botón de reset
4. Leds indicadores

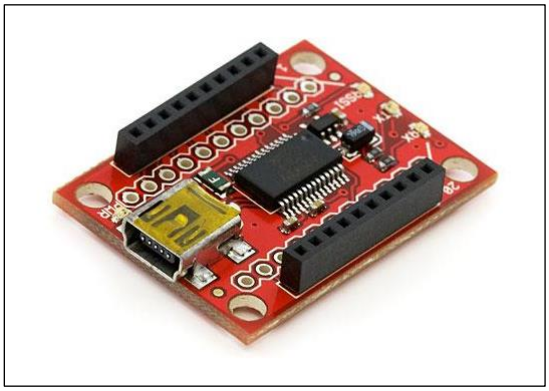


Figura 17. Explorer USB

### 5.6.2.3. Módulo Shield  Xbee Arduino

Los módulos shield xbee- arduino (ver la figura 18), son placas utilizadas en proyectos donde se necesita la comunicación entre el arduino mega y el xbee para él envió de información por  comunicación inalámbrica por medio de  radio frecuencia,  la comunicación xbee evita la conexión por cable a una estación de control donde se va a visualizar los datos  transmitidos para su análisis.

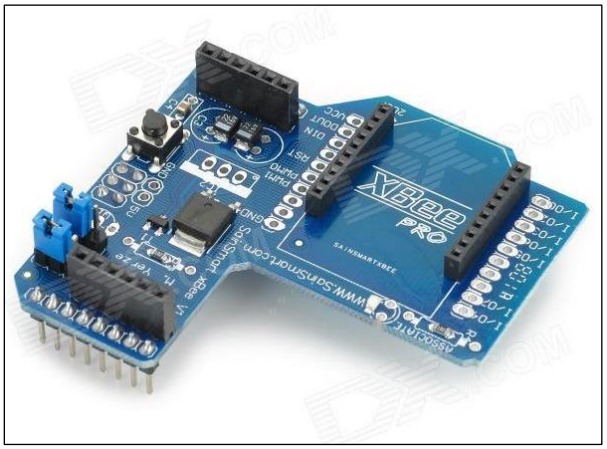



Figura 18. Módulo Shield  Xbee Arduino

## 5.7. Control de robot

Al principio de este capítulo se habló del diseño mecánico y electrónico del robot, explicando con qué dispositivos se realizan los diferentes controles del mismo.

En primer lugar, se procederá a la explicación del control de los motores con los encoders y el control de velocidad, para posteriormente centrarnos en el control del anillo de ultrasonido e infrarrojo.

### 5.7.1. Control de los motores

El control de los motores se desarrolló de manera gradual, debido a la complejidad proyecto no se pudo desarrollar el control de una sola vez, sino que se fueron controlando poco a poco cada uno de los elementos del motor.

Con esto se quiere decir que, para controlar los motores del robot no basta con controlar el motor de corriente continua de manera independiente, es necesario también controlar el driver y los encoders.

En primer lugar, se desarrolló todo el control de forma independiente para cada motor, empezando con el control de los encoders, para después controlar el driver y finalmente controlar la velocidad de cada motor mediante un control PI.

### 5.7.2. Lecturas de los encoders

Los codificadores incrementales, suelen tener cuatro pines en general de los cuales dos de pines son salidas digitales, para el caso de nuestro proyecto solo se conectó la salida del canal A debido a que el movimiento de la plataforma lo necesitamos hacia adelante.

Cuando el encoder gira, produce un pulso de onda cuadrado que se envía por los canales y que representa cuánto gira el eje del encoder. Si se cuentan el número de pulsos generados desde su posición inicial, se podrá conocer cuánto ha girado el encoder. Para determinar la distancia recorrida por la plataforma se debe contar primero los pulsos por vuelta del encoders y se debe tener en cuenta las siguientes variables. En la figura 15 se puede observar cómo se calcula la distancia recorrida a partir de los pulsos del encoders.

- Radio de las llantas = R
- Distancia recorrida = S $S= R*2\pi PPV*C$, (Ec .3)
- Pulsos por vuelta = PPV
- Número de pulsos recorridos= C

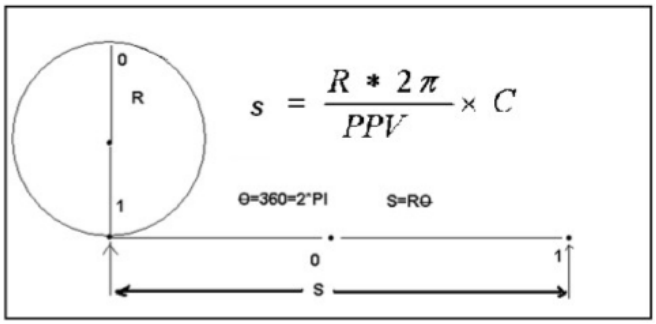



Figura 19. Modelo para calcular la distancia.

La ventaja del encoders steman es que cuenta por vuelta 500 pulsos lo cual lo hace muy preciso. Por lo tanto podemos contar los pulsos utilizando las interrupciones del Arduino mega que para este caso se usó el flanco de bajada en las señales enviadas por el encoders, haciendo que el microprocesador abandone la tarea que estuviera realizando en ese momento para atender la interrupción.

Una vez atendida la interrupción, es decir, una vez se haya actualizado el contador de pulsos, el microcontrolador continúa la tarea que dejó pendiente. El desarrollo del código se encuentra en el anexo xxx, donde se puede observar cómo se usa las interrupciones.

La funciones que se utilizaron para las interrupciones donde se va a ejecutar de manera automática son “doencodersA y doencodersB”.

### 5.7.3. Control del driver del motor

Una vez verificado el funcionamiento de los encoders, se procedió al control de la velocidad del motor. El control del voltaje se realizará a través del PWM del microcontrolador Arduino mega por medio de la función “map”.

El driver L298n tiene seis entradas y cuatro salidas. Las entradas están conectadas directamente al arduino y las salidas al motor. Estas entradas son:

• Entrada de habilitación del movimiento “ENABLE” que trabaja los PWM.

• Los pines de sentido de giro IN1, IN2, IN3 e IN4.

A través de la entrada digital de habilitación se controla los motores mediante el PWM, cuanto mayor sea el valor introducido por este canal, mayor será la velocidad a la que gire la rueda. Este valor estará limitado, ya que el rango para la velocidad del motor es de 0 a 255.

Este control se ha realizado de manera independiente para cada motor, es decir, hay un programa para el motor derecho y otro para el motor izquierdo y ambos se pueden ver en el Anexo . Control del driver.

### 5.7.4. Cinemática diferencial

Una vez conocido el desplazamiento realizado por cada pulso girado, se podrá saber fácilmente el desplazamiento de cada rueda en un periodo de tiempo determinado,
Teniendo en cuenta que en el control de los encoders ya se ha calculado el diferencial de pulsos. De esta forma, la distancia lineal recorrida en un intervalo de tiempo por cada rueda vendrá representada de las siguientes variables como se puede observar en la figura 20.

$$D = \frac{D_D + D_L}{2} \qquad (\text{Ec. } 4)$$

Dónde:

$D_D$= Distancia recorrida de la rueda derecha.

$D_L$= Distancia recorrida de la rueda izquierda.

D=Distancia promedio recorrida del robot.

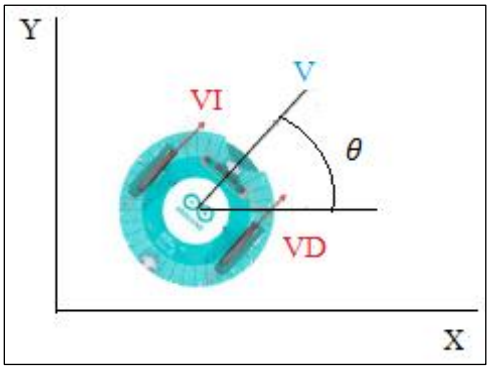



Figura 20. Variables del movimiento del robot diferencial

De forma similar, la velocidad del robot está dada por la siguiente ecuación:

$$V = \frac{V_D + V_I}{2} \qquad (\text{Ec}.5)$$

Dónde:

$V_D$ =Velocidad de la rueda derecha.

$V_I$ =Velocidad de la rueda izquierda.

$V$ =Velocidad promedio del robot.

$X_F$ = Posición horizontal final del robot.

$Y_F$ =Posicion vertical final del robot.

$$\theta = \tan^{-1}\left(\frac{Y_F}{X_F}\right) \qquad (\text{Ec}.6)$$

$X$ = Posición actual horizontal del robot

$Y$ =Posicion actual vertical del robot.

.

$$X = V * \cos\theta \qquad (\text{Ec}.7)$$

$$Y = V * \sin\theta \qquad (\text{Ec}.8)$$

### 5.7.5. Control PI de la velocidad

Después de controlar la velocidad de los motores y su sentido de giro, es necesario mantener estable dicha velocidad. Cuando la plataforma comience a andar por el suelo, es muy importante que ambos motores giren a la velocidad deseada de forma constante, ya que es necesario que la plataforma avance en línea recta. Para que esto pueda suceder, ambas ruedas deberán alcanzar la velocidad establecida rigiéndose por el control PI.

Un controlador PI es un mecanismo de control por realimentación, cuyo diagrama de bloques se podría representar según se observa en el Diagrama 1. Al controlador PI le llega una señal de error, la cual se obtiene comparando el valor real y el valor de referencia. Con este error, el control PI actúa de forma programada para minimizar el error lo más rápido posible.

Este control PI corrige el error mediante la combinación de dos acciones diferentes, acción integral y proporcional.

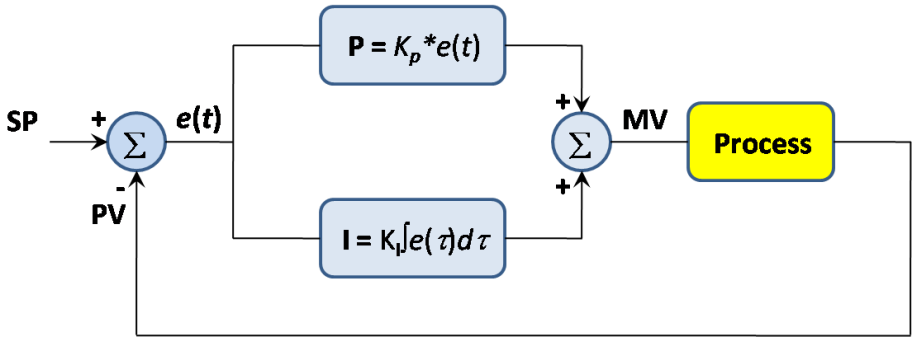



Diagrama 1. Control PI.

Fuente:wikimedia[en línea]. Disponible en web:
<https://upload.wikimedia.org/wikipedia/en/2/26/PI_controller.png>

### 5.7.5.1. Acción proporcional

La acción proporcional es la acción más importante. De hecho, existen controles PID cuya única acción que interviene es la proporcional. Esta acción da como resultado una salida que es proporcional al error, multiplicándose por un valor, llamado ganancia o constante proporcional, Kp.

Partiendo de los parámetros que se observan en el diagrama 1, si sólo actuará la acción proporcional, la salida del controlador sería la que se representa en la siguiente ecuación.

$$\boldsymbol{y(t) = e(t) * k_P} \qquad \text{(Ec . 9)}$$

Gracias a esta acción, conseguiremos que los motores alcancen el valor de referencia rápidamente. Sin embargo, hay que tener cuidado al dar valores a la constante proporcional, ya que si este valor es excesivamente alta, se correrá el riesgo de que haya un sobre-amortiguamiento, es decir, el PI hará que el motor gire a una velocidad excesiva.

En la Figura 21. Se puede observar cómo al aumentar la acción proporcional, la señal alcanza el valor estacionario más rápidamente, sin embargo los valores son demasiado elevados y pueden provocar un sobre-amortiguamiento.

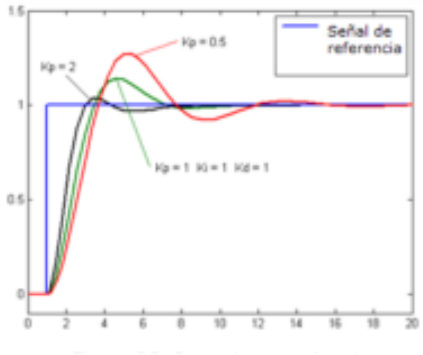



Figura 21. Control proporcional.

### 5.7.5.2. Acción integral

Esta acción da como resultado una salida que es proporcional al error acumulado.

Para poder obtener este resultado será necesario integrar el valor del error durante un tiempo determinado y multiplicarlo por la constante integral, Ki.

Partiendo de los parámetros que se observan en la el diagrama 1, si sólo actuará la acción integral, la salida del controlador sería la que se representa en la siguiente ecuación.

$$\boldsymbol{y(t) = K_i * \int e(t) * dt} \qquad \text{(Ec . 10)}$$

Gracias a esta acción, se conseguirá eliminar el error existente en estado estacionario respecto al valor de referencia. Sin embargo, hay que aplicar esta acción con precaución, ya que se obtiene una respuesta más lenta y el periodo de oscilación aumenta. Lo cual se puede observar en la Figura 22.

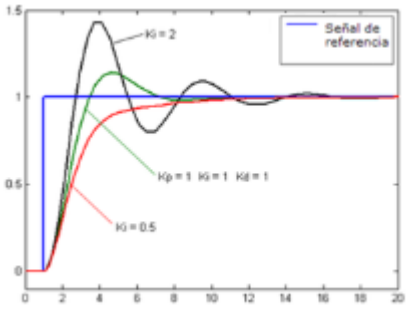



Figura 22. Control integral.

Mediante la combinación de estas acciones, se conseguirá un control adecuado de la velocidad de los motores. A la hora de implementar este control, hay que tener presentes todas las ventajas y desventajas que se pueden producir al aumentar o disminuir los valores de las constantes, las cuales se pueden ver en las tablas Tablas 3 y 4.

| | Ventajas | Desventajas |
|---|---|---|
| Control proporcional | Error en estado estacionario disminuye. El proceso responde más rápidamente. | La sobre-amortiguación y las oscilaciones aumentan. |
| Control integral | Se elimina el error en estado estacionario. | Cuanto menor sea la constante, habrá más oscilaciones. Da inestabilidad. |

Tabla 3. Ventajas y desventajas de los controles PI.

| | $K_p \uparrow$ | $K_i \uparrow$ |
|---|---|---|
| Estabilidad | Disminuye | Disminuye |
| Velocidad | Aumenta | Aumenta |
| Error en estado estacionario | No se elimina | Se elimina |

Tabla 4. Variación de las constantes PI.

### 5.8. Comunicación I2C Arduino Mega maestro - esclavo

La comunicación existente entre el Arduino Mega maestro, el Arduino esclavo y la brújula digital es de tipo I2C.

I2C es un bus de comunicación de tipo multi-maestro, permitiendo que haya varios dispositivos que actúen como maestro y varios dispositivos que actúen como esclavo dentro del mismo bus. En este caso, habrá un maestro, un esclavo y un dispositivo, los cuales serán dos Arduino Mega y la brújula, respectivamente.

El bus I2C está formado por tres líneas:

• Masa, GND. Esta línea debe ser común para todos los dispositivos que estén conectados al bus I2C.

• Datos, SDA. Por esta línea circulan los datos que se envían entre los dispositivos.

• Reloj, SCL. Esta línea sirve de sincronización mediante pulsos de reloj.

Una posible conexión de este tipo de bus se puede observar en la figura 23, en el cual sólo hay conectados un maestro, un esclavo y un dispositivo, pudiendo existir múltiples dispositivos maestro y esclavo conectados al bus.

Al existir una única línea de datos, hay que indicar siempre hacía donde va dirigido cada dato y hay que realizar un control de acceso al bus, puesto que el bus sólo puede transmitir un dato.

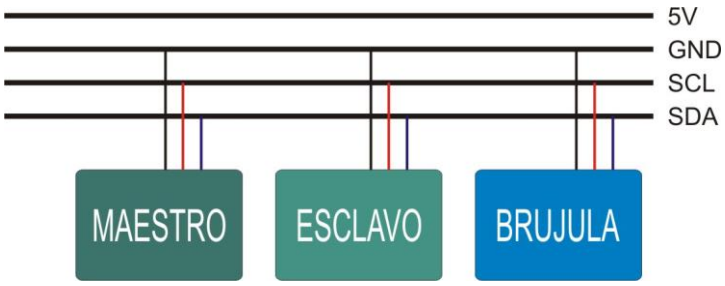



Figura 23. Esquema Comunicación I2C.

Los dispositivos maestro son los únicos que pueden iniciar una comunicación, o lo que es lo mismo, son los únicos que pueden controlar la línea SCL, por tanto, los dispositivos esclavo siempre están a la espera de recibir una petición de datos, para poder enviar información al maestro por la línea SDA.

Si un maestro quiere empezar a comunicarse con otro dispositivo y están las líneas SDA y SCL inactivas, deberá ocupar el bus, indicándoselo al resto de dispositivos conectados al bus activando la línea SDA. Hasta que este dispositivo no libere el bus, ningún otro dispositivo podrá empezar a comunicarse. Este proceso puede se puede observar en la Figura 24. Una vez inicializada la comunicación, el maestro envía un byte formado por 8 bits, (véase en la Figura 25) de los cuales 7 sirven para indicar la dirección del dispositivo con el que se quiere intercambiar información y el octavo bit sirve para indicar el tipo de acción que se quiere realizar, a saber, lectura (nivel alto) o escritura(nivel bajo).

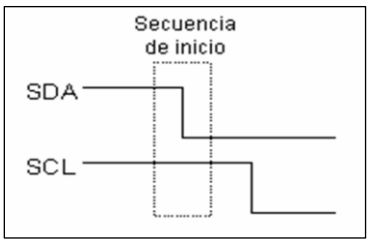



Figura 24. Inicio de comunicación I2C.

Si el esclavo se encuentra en disposición de comunicarse, envía una señal ACK a nivel bajo, indicándole al maestro que puede iniciar la transferencia de datos. En caso contrario, la señal de ACK mandada estaría a nivel alto, y el maestro no podría intercambiar información.

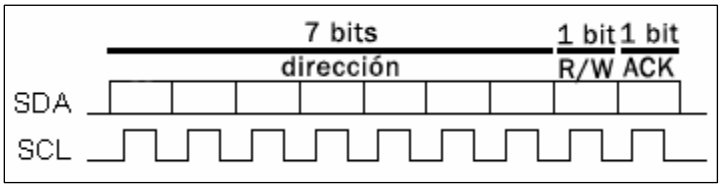



Figura 25. Bits de direccionamiento.

Una vez terminada toda la transmisión de datos, el maestro debe liberar el bus para que otros puedan utilizarlo. Para liberarlo será necesario poner la línea SDA en nivel lógico alto, tal y como se observa en la Figura 26.

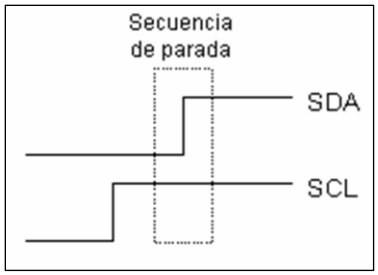



Figura 26. Fin de comunicación I2C.

### 5.8.1. Comunicación I2C en Arduino

Teniendo clara la comunicación I2C, continuamos a la implementación de este tipo de comunicación en el robot PMITO. En primer lugar, hay que realizar las conexiones adecuadas para una correcta comunicación. Como ya se ha visto, es necesario el uso de cuatro líneas, SDA, SCL, 5V y GND. En el Arduino Mega los pines digitales 20 y 21, son los correspondientes a SDA y SDL respectivamente. Dicha conexión se puede ver en la figura 27.

Una vez realizado el cableado, se debe implementar toda la comunicación descrita en los códigos. Para ello, se debe incluir las librerías correspondientes y se define el papel de cada Arduino dentro de esta comunicación.

En este caso, un Arduino Mega actúa como maestro y el otro Arduino Mega como esclavo.

El Arduino Mega maestro, inicia la comunicación en el bucle mediante los respectivos comandos y a su vez intercambiando información.

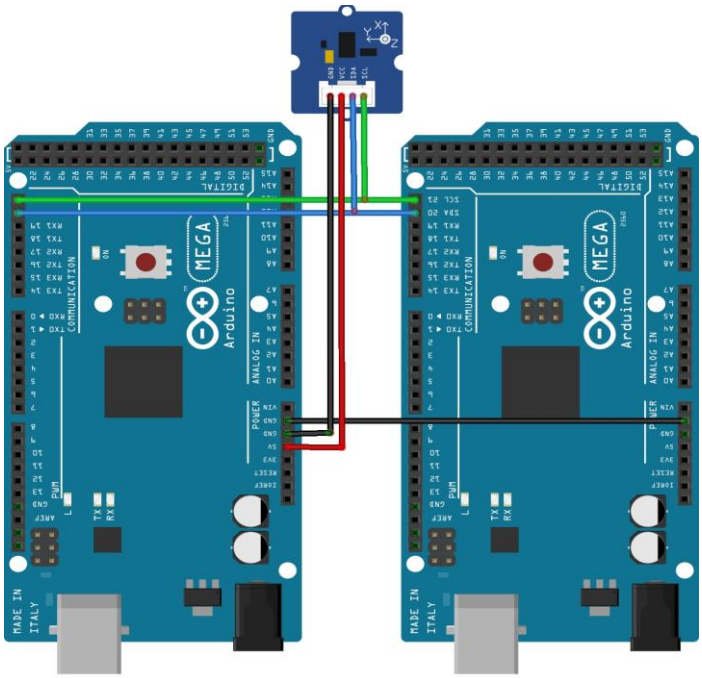


Figura 27. Esquema de conexión de la comunicación I2C.

### 5.8.2. Comunicación Arduino Mega maestro – Ordenador

La comunicación entre Arduino y el ordenador se basa en puertos serie.

Para realizar el proceso de comunicación con el control de alto nivel basta con conectar mediante un cable el microcontrolador con el ordenador, pudiendo de esta manera mandar y recibir información del microcontrolador. Para ello, es necesario un interface, el software de Matlab o Arduino.

En primer lugar, será necesario abrir el puerto serie y fijar la velocidad en baudios para la transmisión de datos en serie. El valor utilizado en este caso es el de 115200 baudios, y esta apertura se realizará en el bucle void setup(), mediante el comando:

Serial.begin(115200);

Abierto el puerto serie, tan sólo queda mandar y recibir datos. Para enviar será necesario imprimir los datos en el puerto serie mediante dos tipos de comandos:

Serial.print(data);

Serial.println(data);

Ambos funcionan de la misma manera, imprimen el dato colocado entre paréntesis por el puerto serie, sin embargo, el segundo comando además de imprimir el dato, imprime un salto de línea y así identificar el fin de una trama de datos.

Para recibir un dato se utilizarán los siguientes comandos:

Serial.available();

Serial.read();

El primer comando devuelve un número entero con el número de caracteres disponibles en el buffer (capacidad máxima de 128 byte), es decir, el número de datos enviados. En el caso de que no se enviará ningún dato, el comando devuelve un 0. En caso contrario, será necesario leer los datos recibido mediante el segundo comando escrito, el cual es capaz de leer un solo byte del puerto serie. Una vez conseguida la comunicación entre el microcontrolador y el ordenador, hay que tener en cuenta que todo el proceso de control del robot PMITO debe comunicarse con un programa realizado en Matlab, el cual sigue un protocolo para enviar y recibir datos.

### 5.9. Control general

Se unificaron todos los controles en dos códigos, uno destinado al Arduino Mega maestro y otro destinado al Arduino Mega esclavo.

El código del Arduino Mega maestro es el descrito en el anexo xxx donde se toma las medidas de los sensores ultrasonidos e infrarrojos, brújula y los datos de la interfaz de

usuario, para poder enviarlos al Arduino Mega esclavo, y a su vez recibir datos del mismo para enviarlos al ordenador para que este reciba los datos en la interfaz de Matlab. Para mayor claridad, se ha representado las descripciones de la conexión para la función del maestro- esclavo se puede observar en el figura 28 y 29.


Figura 28. Esquema conexión de los dispositivos usados al maestro.

El código Arduino Mega esclavo, es el descrito en el anexo xxx donde se encuentra el control PI de la velocidad, control de la brújula y número de pulsos dados para enviarlos al Arduino Mega maestro.

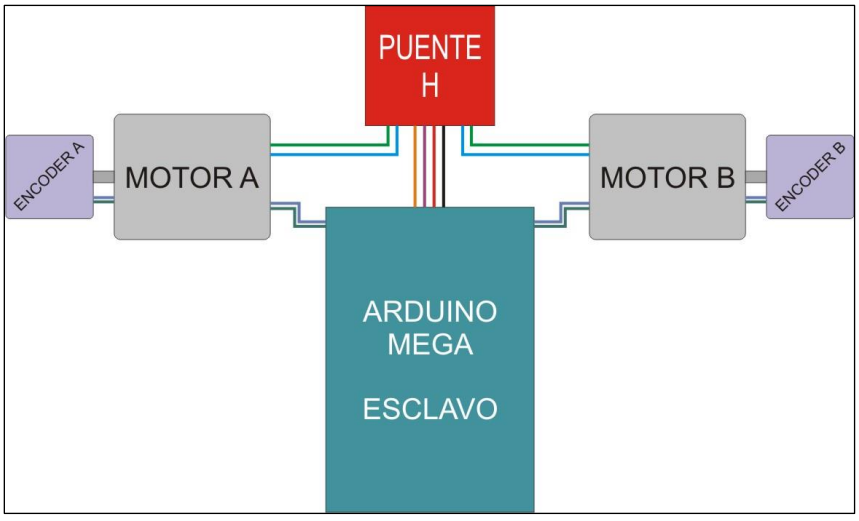



Figura 29. Esquema Conexión de los dispositivos usados al esclavo.

Además de la unificación de todos estos controles, se tuvieron que añadir dos tipos de comunicación:

• Comunicación I2C, para comunicar los dos Arduino Mega.

• Comunicación puerto serie, para comunicar el Arduino Mega con el alto nivel (ordenador).

Teniendo en cuenta cómo se relacionan los dispositivos de control, se ilustran los diagramas de flujo, que comprenden todo el código del proceso de control y comunicación del proyecto.

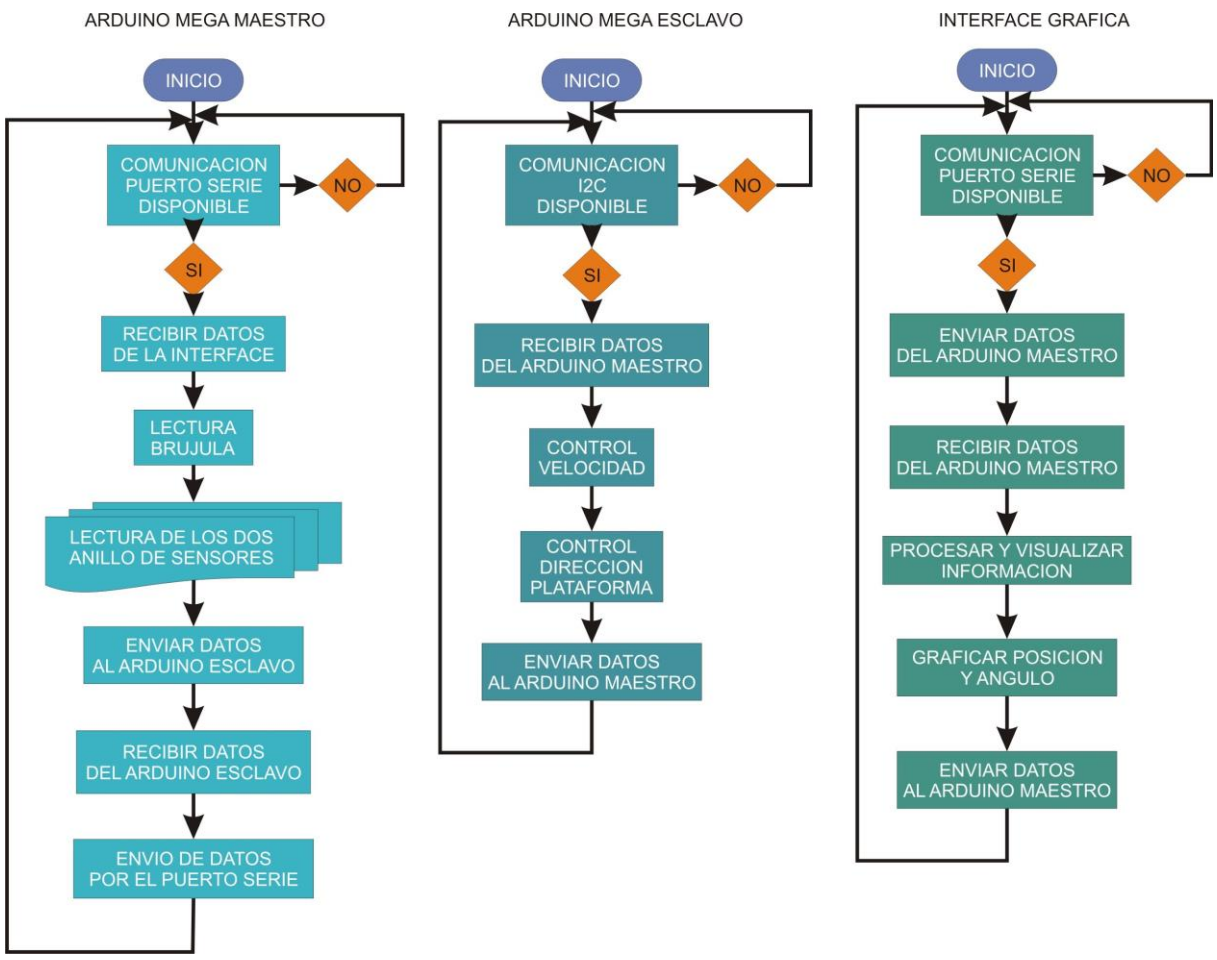



Figura 30. Diagrama de flujo del sistema”PMITO ” en general.

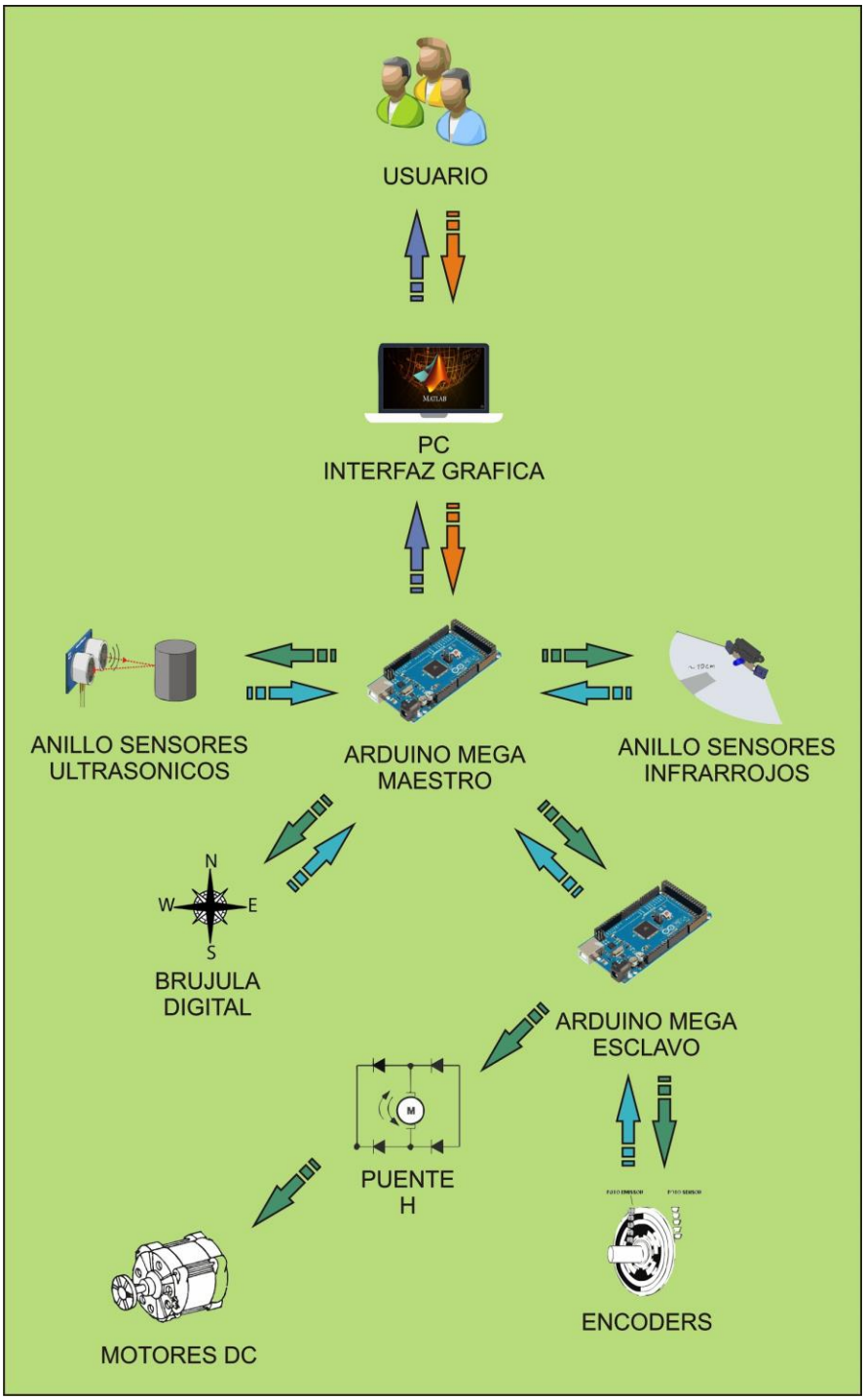



Figura 31. Esquema general de comunicación.

## 5.10. Localización

Para la localización se utiliza la interfaz de usuario de Matlab la cual recoge la información necesaria de la plataforma para así determinar la posición y ángulo en cada instante de tiempo de dicha plataforma, lo cual se puede observar en la Figura 32.

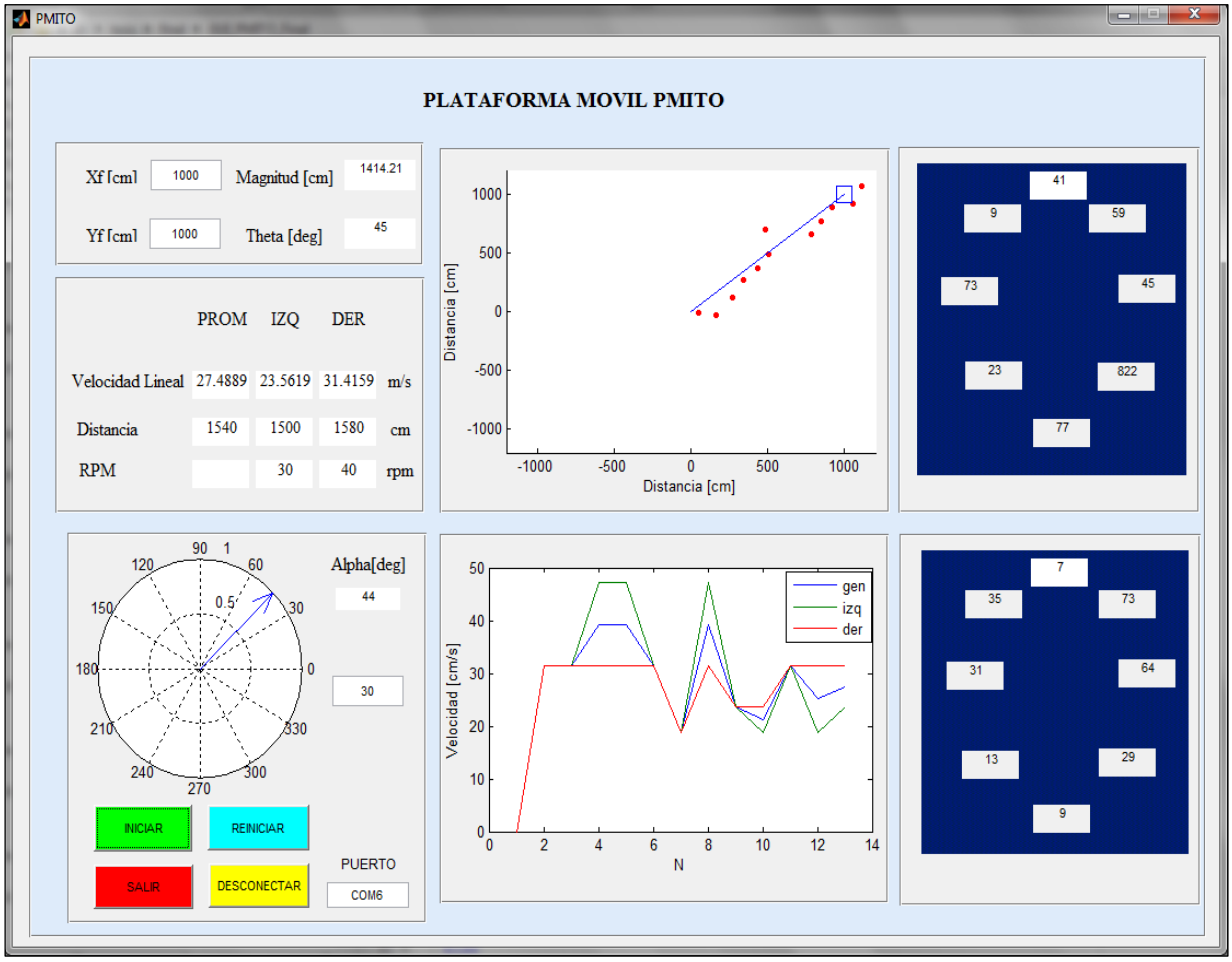



Figura 32. Guide en Matlab para la trayectoria y localización de la plataforma.

## 6. RESULTADOS

Para implementar y calcular las constantes del controlador se hizo pruebas a los motores para determinar en qué valor de PWM giraban, esto se hizo variando el PWM desde cero hasta el valor que se pudo apreciar el arranque en cada motor; para el motor A se obtuvo un valor de 28 y para el motor B un valor de 31 (el rango de PWM es 0-255).

Se hizo pruebas del control PI en el arranque del motor observando cómo se estabiliza la señal controlada, notándose en la Figura 33a, para el control del motor A y en la Figura 33b, para el control del motor B

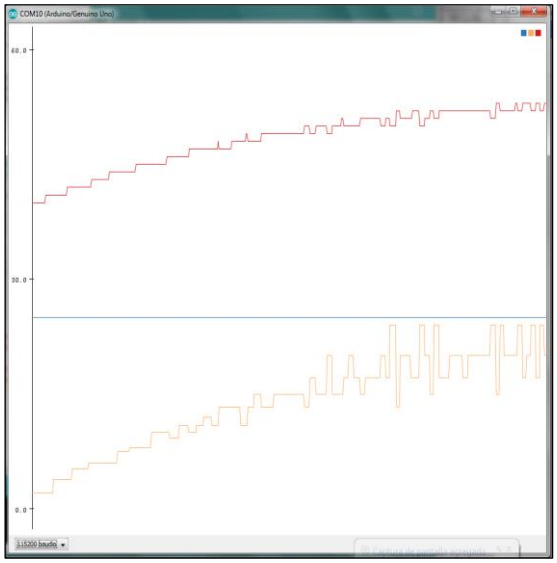


Figura 33a. Control para el Motor A

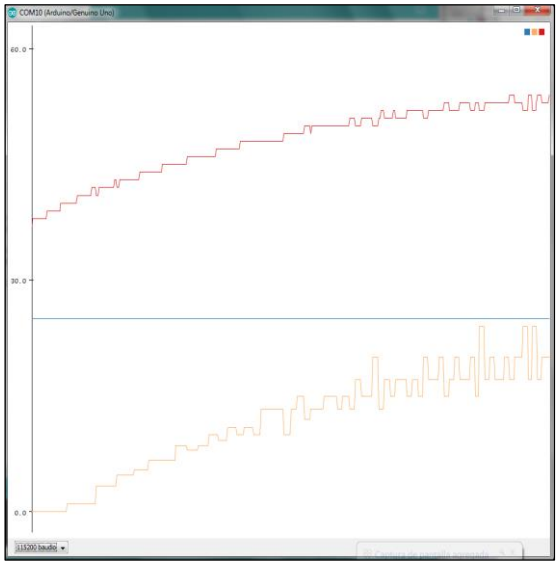


Figura 33b. Control para el Motor B

Luego se trata de frenar la rueda y notamos una rápida respuesta del control reflejada, esta respuesta se puede observar en la Figura 34a; al momento de liberar la rueda se aprecia un aumento de la velocidad siendo corregido por el control, esto lo notamos en la Figura 34b, luego de esta prueba se aprecia en la Figura 34c, una estabilidad en la señal que queremos controlar, después realizamos una última prueba haciendo una perturbación acelerando la rueda y notamos la respuesta del control en la Figura 34d, logrando así una correcta respuesta del control implementado.

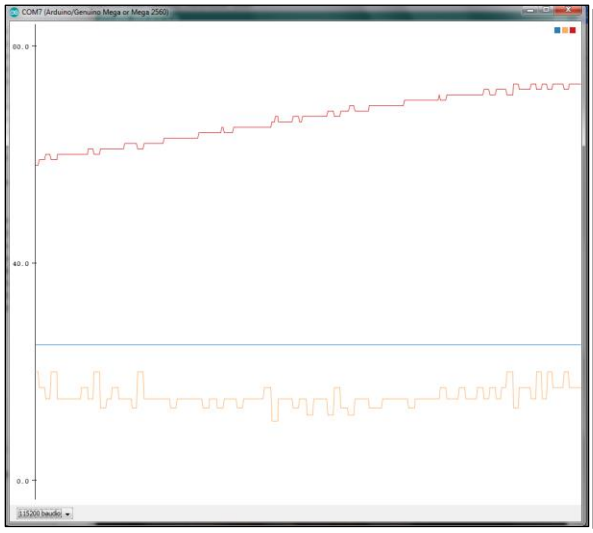


Figura 34a. Frenado en la rueda.

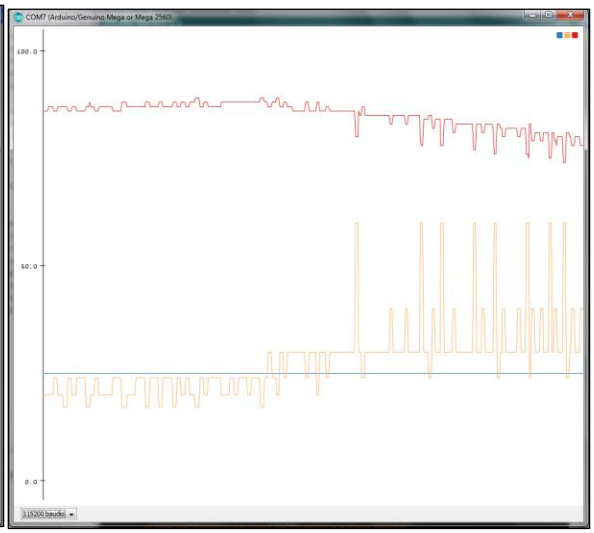


Figura 34b. Liberación de la rueda frenada.

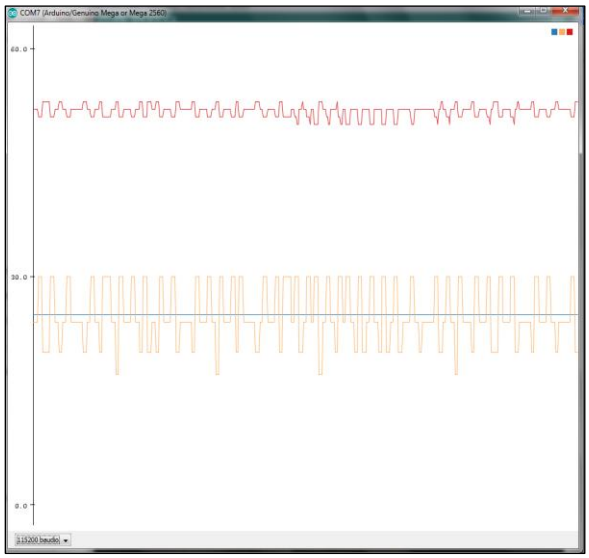


Figura 34c. Señal controlada estable.

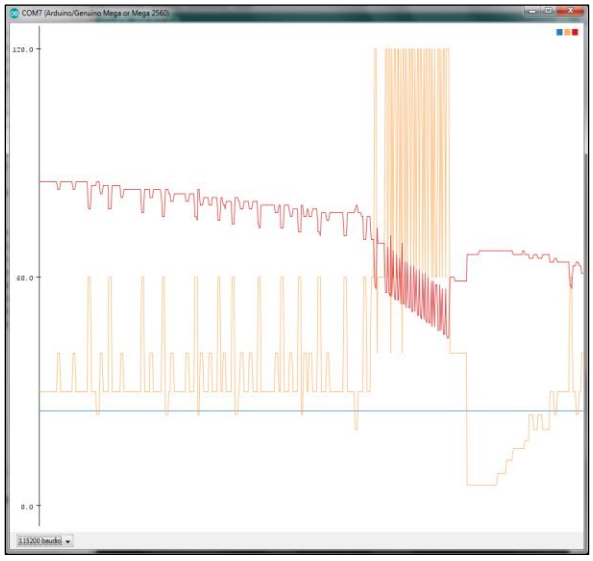


Figura 34d. Perturbación externa en la velocidad.

A continuación se tomaron medidas para cada uno de los sensores tanto infrarrojos como ultrasónicos obteniendo los resultados en el anexo 2 y anexo 3 respectivamente, de estos resultados se escogió la altura de 20 cm en el caso del infrarrojo y 30 cm para el ultrasónico, estas fueron escogidas de acuerdo a la construcción de la plataforma, luego se calculó el error rms para cada sensor teniendo en cuenta la fórmula (Ec. 11) y los valores consignados en la tablas 5 y 6.

$$ERROR\ RMS = \sqrt{\frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n}} \quad (Ec.11)$$

| cm | Sharp 1 | Sharp 2 | Sharp 3 | Sharp 4 | Sharp 5 | Sharp 6 | Sharp 7 | Sharp 8 |
|---|---|---|---|---|---|---|---|---|
| 5 | 4,85 | 4,88 | 5 | 5,01 | 5,12 | 4,72 | 5,08 | 5 |
| 10 | 10,07 | 9,99 | 9,98 | 9,91 | 10,04 | 9,99 | 10,15 | 10,27 |
| 15 | 14,87 | 15,06 | 4,92 | 14,94 | 14,89 | 14,92 | 14,95 | 15 |
| 20 | 19,68 | 20,63 | 20,37 | 20,51 | 19,79 | 20 | 20,45 | 19,94 |
| 25 | 24,98 | 25,42 | 25,47 | 25,27 | 25,4 | 24,69 | 25,72 | 25,02 |
| 30 | 30,44 | 29,26 | 30,48 | 29,19 | 30,18 | 30,05 | 30,22 | 30,74 |
| 35 | 35,28 | 33,27 | 34,65 | 33,36 | 34,79 | 35,8 | 34,86 | 36,89 |
| **E.RMS** | 0,24 | 0,77 | 0,32 | 0,73 | 0,21 | 0,34 | 0,34 | 0,77 |

Tabla 5. Valores y error rms de los sensores infrarrojos a 20 cm de altura.

| cm | HC-SR1 | HC-SR2 | HC-SR3 | HC-SR4 | HC-SR5 | HC-SR6 | HC-SR7 | HC-SR8 |
|---|---|---|---|---|---|---|---|---|
| 10 | 10,07 | 9,65 | 10,9 | 10,4 | 10,7 | 10,3 | 9,6 | 10,1 |
| 20 | 19,98 | 19,8 | 20,0 | 19,8 | 21,0 | 20,8 | 20,3 | 20,2 |
| 30 | 30,02 | 29,7 | 29,8 | 29,9 | 30,7 | 30,7 | 29,7 | 29,7 |
| 40 | 40 | 39,6 | 39,7 | 39,7 | 40,0 | 39,8 | 40,0 | 39,8 |
| 50 | 49,62 | 49,7 | 50,0 | 49,6 | 50,0 | 50,0 | 49,0 | 49,7 |
| 60 | 59,2 | 59,5 | 59,7 | 59,0 | 60,4 | 59,5 | 59,1 | 59,6 |
| 70 | 68,9 | 69,6 | 69,8 | 69,2 | 70,0 | 69,1 | 69,0 | 69,0 |
| 80 | 79.7 | 79,7 | 79,3 | 79,0 | 79,8 | 78,9 | 78,4 | 79,1 |
| **E.RMS** | 0,35 | 0,44 | 0,62 | 0,52 | 0,66 | 0,85 | 0,53 | 0 |

Tabla 6. Valores y error rms de los sensores ultrasónicos a 30 cm de altura.

Por último se integraron todos los módulos estudiados en este documento con el fin obtener una plataforma móvil que siga una trayectoria planteada por el usuario de la forma más correcta posible a la ideal como lo muestra la Figura 35; para tal fin se generaron múltiples trayectorias corrigiendo la velocidad máxima a la que debe ir la plataforma para un correcto funcionamiento.

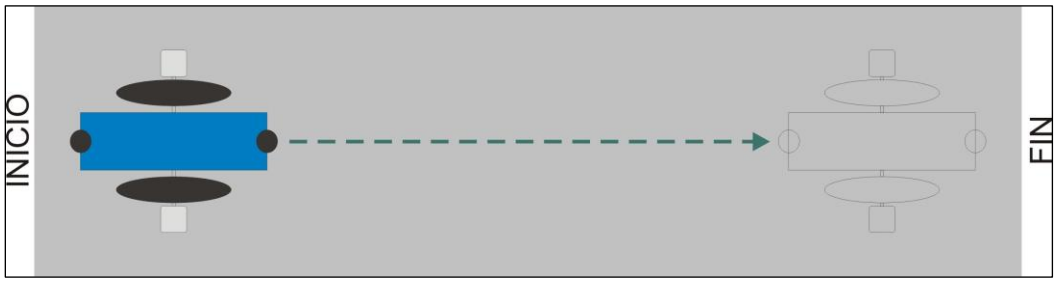



Figura 35. Trayectoria ideal de la plataforma.

Entre los datos ideales y los datos reales se aprecia una dispersión observándola en la Figura 36, esta muestra la diferencia existente entre la trayectoria planteada por el usuario y la recorrida por la plataforma, arrojando un error rms del 6.7% arrojado por el programa al comparar las dos trayectorias,

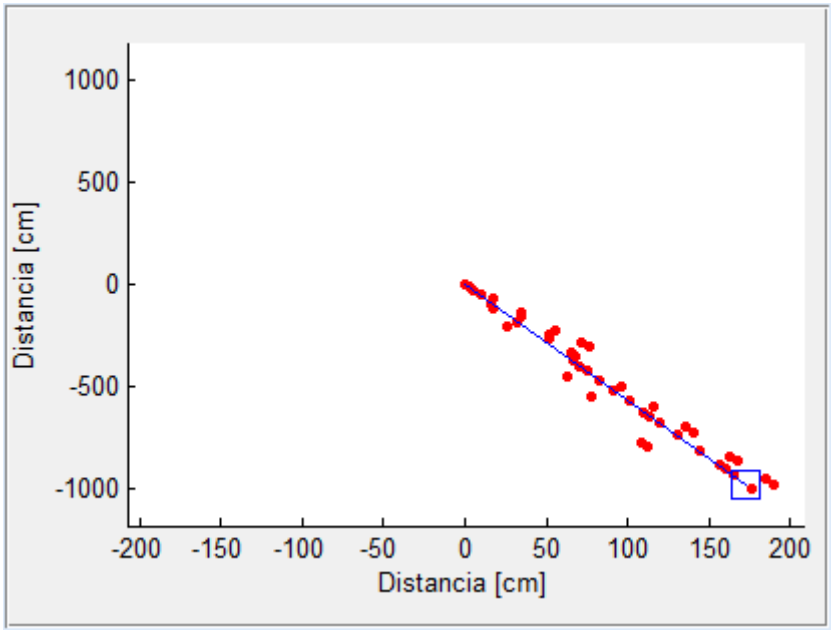



Figura 36. Trayectoria real de la plataforma.

## 7. CONCLUSIONES

- Se puede notar que el control integrado a la velocidad de los motores, es un control robusto contra perturbaciones, siempre y cuando se tengan las ganancias del control proporcional integral debidamente calibradas, además debería implementarse un esquema de control adaptativo para mejorar la funcionalidad de la plataforma sobre diferentes superficies.

- Con la calibración de los sensores se puede observar un error que varía de 0.24 a 0.77 cm para el infrarrojo y de 0.00 a 0.85 cm para el ultrasónico; estos errores se pueden considerar grandes, pero respecto a la plataforma no es relevante ya que para una vuelta completa en las ruedas de esta equivale a 47.1 cm, lo cual se considera como un error aceptable para los estudios de odometría.

- La plataforma móvil junto con el software desarrollado muestra un error rms del 6.7% en la trayectoria ideal de la real, además compara los resultados de los módulos sensoriales con las medidas reales tomadas del entorno lo que demuestra que el proyecto cumple a cabalidad con los objetivos planteados, permitiendo que este proyecto pueda ser útil para otros estudios como levantamiento de mapas de entorno, navegación autónoma en entornos dinámicos implementando algoritmos adecuados para dichos estudios.

## 8. BIBLIOGRAFÍA


[1] Adams. M.D, "*Sensor Modelling, Design and Data Processing for Autonomous Navigation.", World Scientific Publishing, Series in Robotics and Intelligent Systems. Singapore, 1999.*

[2] Agre, Philip E. y Chapman David. *" What are plans for?"*, *Robotics and Autonomous Systems*. No. 6 (1990); p. 17 – 34.

[3] *IEEE Robotics and Automatization Soviet 2001*.

[4] Akbarally, Huzefa y Kleeman, Lindsay. *"A sonar sensor for accurate 3D target localisation and classification" IEEE International conference on robotics and automation*. 1995.

[5] Mataric, Maja. "A distributed model for mobile robot environment-learning and navigation."Cambridge, MA, 1990.

[6] Parker, L. *"Current State of the Art in Dist Distributed Autonomous Mobile Robotic"* ,*Distributed Autonomous Robotic Systems*. Tokyo. Vol 4, (2000); p. 3-12.

[7] Murphy, R.R."*Introduction to All Robotics.",* MIT Press. 2000.

[8] Everett, H.R. y Peters, A K. *" Sensors for Mobile Robots"* , Wellesley, MA,1995.

[9] Feng, L, Borenstein, J y Everett, H.R. *" Where am I? " Sensors and methods for mobile robot positioning.* Universidad de Michigan. 1996.

[10] Adams. M.D. *"Sensor Modelling, Design and Data Processing for Autonomous Navigation." World Scientific Publishing , Series in Robotics and Intelligent Systems. Singapore.* Vol. 13. 1999.

[11] Simó, J.E., *" Una arquitectura basada en motivaciones para el control de robots móviles."*,Tesis Doctoral. Universidad Politécnica de Valencia. 1997.

[12] Brooks, R.A., *"Elephants don't play chess."* , *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back.* 3-15 MIT Press. London. 1991.

[13] Arbib, M.*" Schema Theory, The Handbook of Brain Theory and Neural Networks.",* MIT Press. Cambridge, 1986.

[14] Larkin, Ronald C*." Reactive Robotic Systems" Internal Report.", Georgia Institute of Technology.* Atlanta, Georgia. 1994.

## 9. ANEXOS

Los anexos se encuentran en el archivo adjunto con este documento llamado ANEXOSPMITO.docx